Sumbitted to *Neural Computation*. Preliminary version appeared in Proc. $NIPS*99$

Correctness of belief propagation in Gaussian graphical models of arbitrary topology

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

William T. Freeman
MERL, Mitsubishi Electric Research Labs.
201 Broadway
Cambridge, MA 02139
freeman@merl.com

Abstract

Graphical models, such as Bayesian networks and Markov random fields represent statistical dependencies of variables by a graph. Local "belief propagation" rules of the sort proposed by Pearl [20] are guaranteed to converge to the correct posterior probabilities in singly connected graphs. Recently good performance has been obtained by using these same rules on graphs with loops, a method known as "loopy belief propagation". Perhaps the most dramatic instance is the near Shannon-limit performance of "Turbo codes", whose decoding algorithm is equivalent to loopy propagation.

Except for the case of graphs with a single loop, there has been little theoretical understanding 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" algorithm finds the maximum posterior probability estimate for singly connected networks. We show that, even for non-Gaussian probability distributions, the fixed points of the max-product algorithm in loopy networks are at least local maxima of the posterior probability.

These results motivate using the powerful belief propagation algorithm in a broader class of networks, and help clarify the empirical performance results.

Problems involving probabilistic belief propagation arise in a wide variety of applications, including error correcting codes, speech recognition and image understanding. Typically, a probability distribution is assumed over a set of variables and the task is to infer the values of the unobserved variables given the observed ones. The assumed probability distribution is described using a graphical model [14] — the qualitative aspects of the distribution are specified by a graph structure. The graph may either be directed as in a Bayesian network [20, 12] or undirected as in a Markov Random Field [20, 11].

If the graph is singly connected (i.e. there is only one path between any two given nodes) then there exist efficient local message–passing schemes to calculate the posterior probability of an unobserved variable given the observed variables. Pearl (1988) 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"). However, as Pearl noted, the same algorithm is not guaranteed to converge in multiply connected networks, and even if it does, it will not calculate the correct beliefs [20].

Several groups have recently reported excellent experimental results by running algorithms equivalent to Pearl's algorithm on networks with loops [9, 18, 7]. Perhaps the most dramatic instance of this performance is in an error correcting code scheme known as "Turbo codes" [3]. These codes have been described as "the most exciting and potentially important development in coding theory in many years" [17] and have recently been shown [13, 16] to utilize an algorithm equivalent to belief propagation in a network with loops. Although there is widespread agreement in the coding community that these codes "represent a genuine, and perhaps historic, breakthrough" [17] a theoretical understanding of their performance has yet to be achieved.

Progress in the analysis of loopy belief propagation has been made for the case of networks with a single loop [23, 24, 6, 2]. For these networks, it can be shown that:

- Unless all the compatibilities are deterministic, loopy belief propagation will converge.
- An analytic expression relates the correct marginals to the loopy marginals. The approximation error is related to the convergence rate of the messages — the faster the convergence the more exact the approximation.
- 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* but focus primarily on nodes that describe jointly Gaussian random variables. We give an exact formula that relates 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. The covariance estimates will generally be incorrect but we 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 at least local maxima of the posterior probability of loopy networks. Our results motivate using this powerful algorithm in a broader class of networks.

1 Belief propagation in undirected graphical models

Pearl's original algorithm was described for directed graphs, but in this paper we focus on undirected graphs. Every directed graphical model can be transformed into an undirected graphical model before doing inference (see figure 1). An undirected graphical model (or a Markov Random Field) is a graph in which the nodes represent variables and arcs represents compatibility constraints between them. Assuming all probabilities are nonzero, the Hammersley-Clifford theorem (e.g. [20]) guarantees that the probability distribution will factorize into a product of functions of the maximal cliques of the graph.

Denoting by x the values of all unobserved variables in the graph, the factorization has the form:

$$P(x) = \prod_c \Psi_c(x_c) \tag{1}$$

where x_c is a subset of x that form a clique in the graph and Ψ_c is the potential function for the clique.

We will assume, without loss of generality, that each x_i node has a corresponding y_i node that is connected only to x_i.

Thus:

$$P(x, y) = \prod_c \Psi_c(x_c) \prod_i \Psi_{ii}(x_i, y_i) \tag{2}$$

The restriction that all the y_i variables are observed and none of the x_i variables are is just to make the notation simple — $\Psi_{ii}(x_i, y_i)$ may be independent of y_i (equivalent to y_i being unobserved) or $\Psi_{ii}(x_i, y_i)$ may be $\delta(x_i - x_o)$ (equivalent to x_i being observed, with value x_o).

In describing and analyzing belief propagation we assume the graphical model has been preprocessed so that all the cliques consist of pairs of units. Any graphical model can be converted into this form before doing inference through a suitable clustering of nodes into large nodes [24]. Figure 1 shows an example — a Bayesian network is converted into an MRF in which all the cliques are pairs of units.

Equation 2 becomes

$$P(x, y) = \prod_{i,j} \Psi_{ij}(x_i, x_j) \prod_i \Psi_{ii}(x_i, y_i) \tag{3}$$

where the first product is over connected pairs of nodes.

By preprocessing the graph into one with pairwise cliques, the description and the analysis of belief propagation becomes simpler. For completeness, we review the belief propagation scheme used in [24].

At every iteration, each node sends a (different) message to each of its neighbors and receives a message from each neighbor. Let x_i and x_j be two neighboring nodes

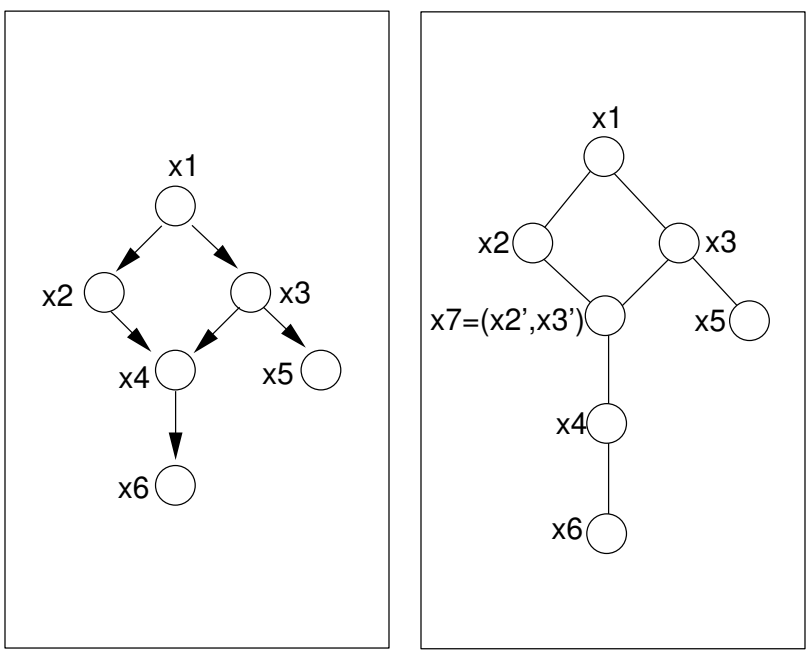

Figure 1: Any Bayesian network can be converted into an undirected graph with pairwise cliques by adding cluster nodes for all parents that share a common child. **a.** A Bayesian network. **b.** The corresponding undirected graph with pairwise cliques. A cluster node for (B, C) has been added. The potentials can be set so that the joint probability in the undirected network is identical to that in the Bayesian network. In this case the update rules presented in this paper reduce to Pearl's propagation rules in the original Bayesian network [24].

in the graph. We denote by $m_{ij}(x_j)$ the message that node x_i sends to node x_j, by $m_{ii}(x_i)$ the message that y_i sends to x_i, and by $b_i(x_i)$) the belief at node x_i.

The belief update (or "sum-product" update) rules are:

$$m_{ij}(x_j) \leftarrow \alpha \int_{x_i} \Psi_{ij}(x_i, x_j) m_{ii}(x_i) \prod_{x_k \in N(x_i) \backslash x_j} m_{ki}(x_i) \tag{4}$$

$$b_i(x_i) \leftarrow \alpha m_{ii}(x_i) \prod_{x_k \in N(x_i)} m_{ki}(x_i) \tag{5}$$

where α denotes a normalization constant and $N(x_i) \backslash x_j$ means all nodes neighboring x_i, except x_j.

The procedure is initialized with all message vectors set to constant functions. Observed nodes do not receive messages and they always transmit the same vector—if y_i is observed to have value y^* then $m_{ii}(x_i) = \Psi_{ii}(x_i, y^*)$. The normalization of m_{ij} in equation 4 is not necessary–whether or not the message are normalized, the belief b_i will be identical. However, normalizing the messages avoids numerical underflow and adds to the stability of the algorithm. We assume throughout this paper that all nodes simultaneously update their messages in parallel.

It is easy to show that for singly connected graphs these updates will converge in a number of iterations equal to the diameter of the graph and the beliefs are guaranteed to give the correct posterior marginals: $b_i(x_i) = Prob(X_i = x_i|Y)$ where Y denotes the set of observed variables.

This message passing scheme is equivalent to Pearl's belief propagation in *directed* graphs of arbitrary clique size — for every message passed in this scheme there exists a corresponding message in Pearl's algorithm when the directed graph is converted to an undirected graph with pairwise cliques [24]. For particular graphs with particular settings of the potentials, Equs. 4–5 yield other well-known Bayesian inference algorithms, such as the forward-backward algorithm in Hidden Markov Models, the Kalman Filter and even the Fast Fourier Transform [1, 13].

A related algorithm, "max-product", changes the integration in equation 4 to a maximization. This message-passing is equivalent to Pearl's "belief revision" algorithm in directed graphs. For particular graphs with particular settings of the potentials, the max-product algorithm is equivalent to the Viterbi algorithm for hidden Markov models, and concurrent dynamic programming. We define the max-product assignment at each node to be the value that maximizes its belief (assuming a unique maximizing value exists). For singly connected graphs, the max-product assignment is guaranteed to give the MAP assignment.

1.1 Gaussian Markov Random Fields

A Gaussian MRF (GMRF) is an MRF in which the joint distribution is Gaussian. We assume, without loss of generality, that the joint mean is zero (the means can be added-in later), so the joint distribution of $z = \begin{pmatrix} x \\ y \end{pmatrix}$ is given by:

$$Prob(z) = \alpha e^{-\frac{1}{2} z^T V z} \tag{6}$$

where α is a normalization constant and V has the following structure:

$$V = \begin{pmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{pmatrix} \tag{7}$$

It is straightforward to write the inverse covariance matrix describing the GMRF which respects the statistical dependencies within the graphical model [4].

If the graph has been preprocessed such that the maximal cliques are pairwise cliques then the joint distribution must have a representation as a product of pairwise potentials. Thus there exist matrices V_{ij}, one corresponding to each connected pair of nodes and matrices V_{ii} corresponding to each node such that:

$$Prob(z) = \alpha \prod_{i,j} e^{-\frac{1}{2}(x_i x_j) V_{ij} (x_i x_j)^T} \prod_i e^{-\frac{1}{2}(x_i y_i) V_{ii} (x_i y_i)^T} \tag{8}$$

where the first product is again over connected pairs of nodes.

Note that the decomposition of V into V_{ij}, V_{ii} is not unique. For scalar nodes, any set of V_{ij}, V_{ii} that satisfy the following constraints are valid:

$$V_{xy}(i,i) = V_{ii}(1,2) \tag{9}$$

$$V_{xx}(i,j) = V_{ij}(1,2) \tag{10}$$

$$V_{xx}(i,i) = V_{ii}(1,1) + \sum_{x_j \in N(x_i)} V_{ij}(1,1) \tag{11}$$

1.2 Exact inference in Gaussian MRFs

Writing out the exponent of Eq. 6 and completing the square shows that the mean μ of x, given y, is a solution to:

$$V_{xx}\mu = -V_{xy}y \tag{12}$$

and the covariance matrix $C_{x|y}$ of x given y is:

$$C_{x|y} = V_{xx}^{-1} \tag{13}$$

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 $C_{x_i|y}(i)$.

1.3 Belief Propagation for Gaussian MRFs

Belief propagation in Gaussian MRFs gives simpler update formulas than the general case (Eqs. 4 and 5). The messages and the beliefs are all Gaussians and the updates can be written directly in terms of the means and inverse covariance matrices. Each node sends and receives a mean vector and inverse covariance matrix to and from each neighbor, in general, each different.

To explicitly write the updates in terms of means and covariances, we denote by μ_{ij} the mean of the message that node x_i sends to node x_j and by P_{ij} the precision or

inverse covariance matrix that node x_i sends to node x_j. Similarly, we denote by μ_i the mean of the belief at node x_i and by P_i the inverse covariance matrix of the belief. As before, we use μ_{ii}, P_{ii} for the message that y_i sends to x_i. We partition the matrix V_{ij} into

$$V_{ij} = \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} \tag{14}$$

The message update rules are:

$$P_{ij} \leftarrow c - b(a + P_0)^{-1} b^T \tag{15}$$

$$\mu_{ij} \leftarrow -P_{ij}^{-1} b (a + P_0)^{-1} P_0 \mu_0 \tag{16}$$

with:

$$P_0 = P_{ii} + \sum_{x_k \in N(x_i) \setminus x_j} P_{ki} \tag{17}$$

$$\mu_0 = P_0^{-1} \left(P_{ii}\mu_{ii} + \sum_{x_k \in N(x_i) \setminus x_j} P_{ki}\mu_{ki} \right) \tag{18}$$

The beliefs are given by:

$$P_i \leftarrow P_{ii} + \sum_{x_k \in N(x_i)} P_{ki} \tag{19}$$

$$\mu_i \leftarrow P_i^{-1} \left(P_{ii}\mu_{ii} + \sum_{x_k \in N(x_i)} P_{ki}\mu_{ki} \right) \tag{20}$$

P_{ii}, μ_{ii} are computed from equations 15 with $\mu_0 = y^*$ and $P_0 = \infty I$.

We can now state the main question of this paper. What is the relationship between the true posterior means and covariances (calculated using Eq. 12) and the belief propagation means and covariances (calculated using the belief propagation rules Eqs. 15–20) ?

2 Dynamics of Belief Propagation

To compare the correct posteriors and the loopy beliefs, we construct an unwrapped tree. The unwrapped tree is the graphical model that the loopy belief propagation is solving exactly when applying the belief propagation rules in a loopy network [10, 25, 24]. In error-correcting codes, the unwrapped tree is referred to as the "computation tree" — it is based on the idea that the computation of a message sent by a node at time t depends on messages it received from its neighbors at time $t-1$ and those messages depend on the messages the neighbors received at time $t-2$ etc.

To construct the topology of the unwrapped tree, set an arbitrary node, say x_1, to be the root node and then iterate the following procedure t times:

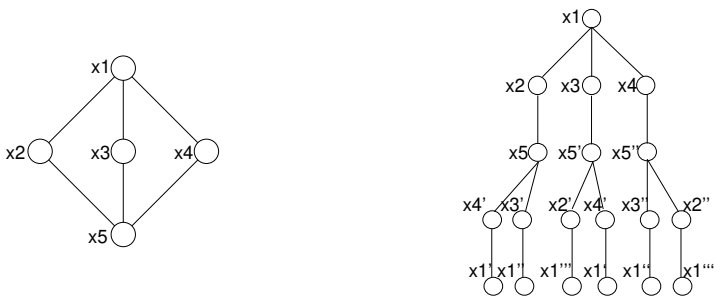

Figure 2: **Left:** A Markov network with multiple loops. **Right:** The unwrapped network corresponding to this structure. The unwrapped networks are constructed by replicating the potentials $\Psi_{ij}(x_i, x_j)$ and observations y_i while preserving the local connectivity of the loopy network. They are constructed so that the messages received by node A after t iterations in the loopy network are equivalent to those that would be received by A in the unwrapped network. An observed node, y_i, not shown, is connected to each depicted node.

- Find all leaves of the tree (start with the root).
- For each leaf, find all k nodes in the loopy graph that neighbor the node corresponding to this leaf.
- Add $k-1$ nodes as children to each leaf, corresponding to all neighbors except the parent node.

The potential matrices and observations for each node in the unwrapped network are copied from the corresponding nodes in the loopy graph. Each node in the loopy graph will have a different unwrapped tree with that node at the root.

Figure 2 shows an unwrapped tree around node A for the diamond shaped graph on the left. Each node has a shaded observed node attached to it that is not shown for clarity. Since belief propagation is exact for the unwrapped tree, we can calculate the beliefs in the unwrapped tree by using the marginalization formulae for Gaussians.

We use $\tilde{}$ 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 scanned in this fashion. Similarly, we denote by $\tilde{y}$ the observed nodes scanned in the same order. As before, $\tilde{z} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$ To simplify the notation, we assume throughout this section that all nodes are scalar valued. In section 4.2 we generalize the analysis to vector valued nodes.

The basic idea behind our analysis is to relate the wrapped and unwrapped inverse covariance matrices. From equation 11 all elements $\tilde{V}_{xy}(i', j')$ and $\tilde{y}(i')$ are copies of the corresponding elements $V_{xy}(i, j)$ and $y(i)$ (where $\tilde{x}_{i'}, \tilde{x}_{j'}$ are replicas of x_i,x_j). Also, all elements $\tilde{V}_{xx}(i', j')$ are copies of $V_{xx}(i, j)$ and elements $\tilde{V}_{xx}(i', i')$ for *non-leaf* nodes are replicas of $V_{xx}(i, i)$. However, the elements $\tilde{V}_{xx}(i', i')$ for the leaf nodes are *not* copies of $V_{xx}(i, i)$ because the leaf nodes are missing some neighbors.

Intuitively, we might expect that if *all* the equations that $\tilde{\mu}$ satisfies are copies of the

equations that μ satisfies, then simply creating $\tilde{\mu}$ by many copies of μ would give a valid solution in the unwrapped network. However, because some of the equations are not copies, this intuition does not explain why the means are exact in Gaussian networks.

An additional intuition, that we formalize below, is that the influence of the non-copied equations (those at the leaf nodes) decreases with additional iterations. As the number of iterations is increased, the distance between the leaf nodes and the root node increases and their influence on the root node decreases. When their influence goes to zero, the mean at the root node is exact.

Although the elements $V_{xx}(i', j')$ are copies of $V_{xx}(i, j)$ for the non-leaf nodes, the matrix $\tilde{V}_{xx}$ is not simply a block replication of V_{xx}. The system of equations that defines $\tilde{\mu}$ is a coupled system of equations. Hence the variance at the root node $\tilde{V}_{xx}^{-1}(1, 1)$ differs from the correct variance $V_{xx}^{-1}(1, 1)$.

In the following section we prove the following four claims regarding the dynamics of loopy belief propagation in Gaussian graphical models.

Assume, without loss of generality, that the root node is x_1. Let $\tilde{\mu}(1)$ and $\tilde{\sigma}^2(1)$ be the conditional mean and variance at node 1 after t iterations of loopy propagation. Let $\mu(1)$ and $\sigma^2(1)$ be the correct conditional mean and variance of node 1. Let $\tilde{C}_{x_1|y}$ be the conditional correlation of the root node with all other nodes in the unwrapped tree then:

Claim 1:

$$\tilde{\mu}(1) = \mu(1) + \tilde{C}_{x_1|y} r \tag{21}$$

where r is a vector that is zero everywhere but the last L components (corresponding to the leaf nodes).

Claim 2:

$$\tilde{\sigma}^2(1) = \sigma^2(1) + \tilde{C}_{x_1|y} r_1 - \tilde{C}_{x_1|y} r_2 \tag{22}$$

where r_1 is a vector that is zero everywhere but the last L components and r_2 is equal to 1 for all components corresponding to non-root nodes in the unwrapped tree that reference x_1. All other components of r_2 are zero.

Claim 3: If the conditional correlation between the root node and the leaf nodes decreases rapidly enough then (1) belief propagation converges (2) the belief propagation means are exact and (3) the belief propagation variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all $\tilde{x}_j$ that are replicas of x_1.

Claim 4: Assume all V_{ij} are diagonally dominant then: (1) belief propagation converges (2) the belief propagation means are exact and (3) the belief propagation variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all $\tilde{x}_j$ that are replicas of x_1.

To obtain intuition, Fig. 3 shows $\tilde{C}_{x_1|y}$ for the diamond figure in Fig. 2. We generated random potential functions and observations for the loopy diamond figure and calculated the conditional correlations in the unwrapped network. 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

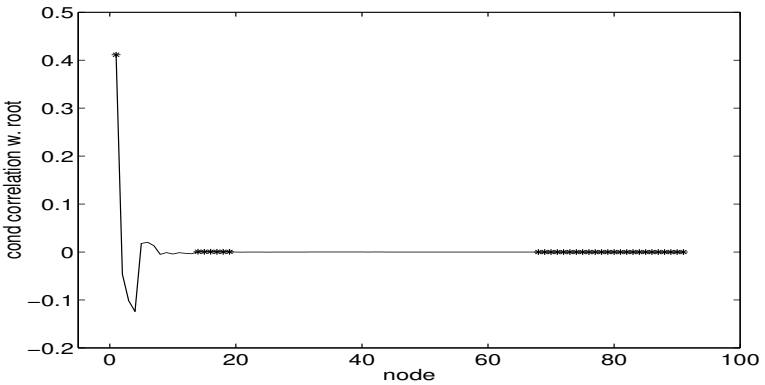

Figure 3: The conditional correlation between the root node and all other nodes in the unwrapped tree for the diamond figure after 7 iterations. Potentials were chosen randomly. Nodes are presented in breadth first order so the last elements are the correlations between the root node and the leaf nodes. It can be proven that if this correlation goes to zero then (1) belief propagation converges (2) the loopy means are exact and (3) the loopy variances equal the correct variances minus the summed conditional correlation of the root node and all other nodes that are replicas of the same loopy node. Symbols plotted with a star denote correlations with nodes that correspond to the node A in the loopy graph. It can be proven that the sum of these correlations gives the correct variance of node A while loopy propagation uses only the first correlation.

tree increases and the conditional correlation between the leaf nodes and the root node decreases.

From equations 21–22 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 belief propagation variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all $\tilde{x}_j$ that are replicas of x_1. In practice the conditional correlations will not actually be equal to zero for any finite unwrapping so claim 3 states this more precisely. Claim 4 gives sufficient conditions, in terms of the V_{ij} matrices for the conditional correlation to decrease rapidly enough.

How wrong will the variances be? The term $\tilde{C}_{x_1|y} r_2$ in Eq. 22 is simply the sum of many components of $\tilde{C}_{x_1|y}$. Figure 3 shows these components. The correct variance is the sum of all the components while the loopy variance approximates this sum with the first (and dominant) term.

Note that when the conditional correlation decreases rapidly to zero two things happen. First, the convergence is faster (because $\tilde{C}_{x_1|y} r_1$ approaches zero faster). Second, the approximation error of the variances is smaller (because $\tilde{C}_{x_1|y} r_2$ is smaller). Thus, as in the single loop case, we find that quick convergence is correlated with good approximation.

2.1 Relation of loopy and unwrapped quantities

The proof of the claims relies on the relationship between the elements of y, V_{xy} and V_{xx} with their unwrapped quantities, described below.

Each node in $\tilde{x}$ corresponds to a node in the original loopy network. Let O be a matrix that defines this correspondence. $O(i,j) = 1$ if $\tilde{x}_i$ corresponds to x_j and zero otherwise. Thus, in figure 2, ordering the nodes alphabetically, the first rows of O are:

$$O = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{23}$$

Using O we can formalize the relationship between the unwrapped quantities and the original ones. The simplest one is $\tilde{y}$, that only contains replicas of the original y:

$$\tilde{y} = Oy \tag{24}$$

Since every x_i is connected to a y_i, V_{xy} and $\tilde{V}_{xy}$ are zero everywhere but along their diagonals (the block diagonals, for vector valued variables). The diagonal elements of $\tilde{V}_{xy}$ are simply replications of V_{xy} hence:

$$\tilde{V}_{xy}O = OV_{xy} \tag{25}$$

$\tilde{V}_{xx}$ also contains the elements of the original V_{xx} but here special care needs to be taken. Note that by construction, every node in the interior of the unwrapped tree has exactly the same statistical relationship with its neighbors as with the corresponding node in the loopy graph. If a node in the loopy graph has k neighbors then a node in the unwrapped tree will have, by construction, one parent and $k-1$ children. The leaf nodes in the unwrapped tree, however, will be missing the $k-1$ children and hence will not have the same number of neighbors. Thus, for all nodes $\tilde{x}_i, \tilde{x}_j$ that are not leaf nodes, $\tilde{V}_{xx}(i,j)$ is a copy of the corresponding $V_{xx}(k,l)$, where unwrapped nodes i and j refer to loopy nodes k and l, respectively.

Therefore:

$$\tilde{V}_{xx}O + E = OV_{xx} \tag{26}$$

where E is an error matrix. E is zero for all non-leaf nodes so the first $N - L$ rows of E are zero.

2.2 Proof of claim 1

The marginalization equation for the unwrapped problem gives:

$$\tilde{V}_{xx}\tilde{\mu} = -\tilde{V}_{xy}\tilde{y} \tag{27}$$

Substituting Eqs. 24 and 25, relating loopy and unwrapped network quantities, into Eq. 27, for the unwrapped posterior mean, gives:

$$\tilde{V}_{xx}\tilde{\mu} = -OV_{xy}y \tag{28}$$

For the true means, μ, of the loopy network, we have

$$V_{xx}\mu = -V_{xy}y \tag{29}$$

To relate that to the means of the unwrapped network, we left-multiply by O:

$$OV_{xx}\mu = -OV_{xy}y. \tag{30}$$

Using Eq. 26, relating V_{xx} to $\tilde{V}_{xx}$, we have

$$\tilde{V}_{xx}O\mu + E\mu = -OV_{xy}y \tag{31}$$

Comparing Eqs. 31 and 28 gives

$$\tilde{V}_{xx}O\mu + E\mu = \tilde{V}_{xx}\tilde{\mu} \tag{32}$$

or:

$$\tilde{\mu} = O\mu + \tilde{V}_{xx}^{-1}E\mu. \tag{33}$$

Using Eq. 13

$$\tilde{\mu} = O\mu + \tilde{C}_{x|y}E\mu. \tag{34}$$

The left and right hand sides of equation 34 are column vectors. We take the first component of both sides and get:

$$\tilde{\mu}(1) = \mu(1) + \tilde{C}_{x_1|y}E\mu \tag{35}$$

Since E is zero in the first $N-L$ rows, $E\mu$ is zero in the first $N-L$ components. □

2.3 Proof of claim 2

From Eq. 13,

$$V_{xx}C_{x|y} = I. \tag{36}$$

Taking the first column of this equation gives:

$$V_{xx}C_{x_1|y}^T = e_1 \tag{37}$$

where $e_1(1) = 1, e_1(j > 1) = 0$.

Using the same strategy as in the previous proof, we left multiply by O:

$$OV_{xx}C_{x_1|y}^T = Oe_1 \tag{38}$$

and similarly we substitute equation 26:

$$\tilde{V}_{xx}OC_{x_1|y}^T + EC_{x_1|y}^T = Oe_1 \tag{39}$$

The analog of equation 37 in the unwrapped problem is:

$$\tilde{V}_{xx}\tilde{C}^T_{x_1|y} = \tilde{e}_1 \tag{40}$$

where $\tilde{e}_1(1) = 1, \tilde{e}_1(j > 1) = 0$.

Subtracting Eqs. 39 and 40 and rearranging terms gives:

$$\tilde{C}_{x_1|y} = OC^T_{x_1|y} + \tilde{V}^{-1}_{xx}EC^T_{x_1|y} + \tilde{V}^{-1}_{xx}(\tilde{e}_1 - Oe_1) \tag{41}$$

Again, we take the first row of both sides of equation 41 and use the fact that the first row of $\tilde{V}^{-1}_{xx}$ is $\tilde{C}_{x_1|y}$ to obtain:

$$\tilde{\sigma}^2(1) = \sigma^2(1) + \tilde{C}_{x_1|y}EC^T_{x_1|y} + \tilde{C}_{x_1|y}(\tilde{e}_1 - Oe_1) \tag{42}$$

Again, since E is zero in the first N_L rows, $EC_{x_1|y}$ is zero in the first $N - L$ components. □

2.4 Proof of claim 3

Here we need to define what we mean by "rapidly enough". We restate the claim precisely.

Suppose for every ϵ there exists a t_ϵ such that for all $t > t_\epsilon$ $|\tilde{C}_{x_1|y}r| < \epsilon \max_i |r(i)|$ for any vector r that is nonzero only in the last L components (those corresponding to the leaf nodes). In this case, (1) belief propagation converges (2) the means are exact and (3) the variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all non-root $\tilde{x}_j$ that are replicas of x_1

This claim follows from the first two claims. The only thing to show is that $E\mu$ and $EC_{x_1|y}$ are bounded for all iterations. This is true because the rows of E are bounded and $\mu, C_{x_1|y}$ do not depend on the iteration. □

2.5 Proof of claim 4:

The proof is based on the following lemma[1].

Conditional correlation lemma: Assume $P(x, y) = \alpha e^{-\frac{1}{2}z^T V z}$ with z, V as in equations 6–7. Let r be an arbitrary vector, then $C_{x|y}r$ can be calculated by (1) modifying the joint probability so that $V_{xy} = -I$ (2) setting the observations $y = r$ and (3) calculating the posterior *means* of x given this y in the modified joint.

Proof: This follows directly from equations 12,13. □

Using this lemma, we can give sufficient conditions for convergence in terms of the potentials of the loopy network.

Claim 4: Assume all V_{ij} are diagonally dominant (i.e. $|V_ij(k, l)| < V_{ij}(k, k)$) and all $V_{ii}(1, 1) > 0$ then: (1) belief propagation converges (2) the belief propagation means are exact and (3) the belief propagation variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all $\tilde{x}_j$ that are replicas of x_1.

[1] We thank Andrew Ng for suggesting this proof.

Proof: The proof is based on claim 3 and the conditional correlation lemma. From claim 3 we know that it is sufficient to show that for any ϵ and r that is nonzero only at the leaves, $|\tilde{C}_{x_1|y}r| < \epsilon \max_i |r(i)|$ for sufficiently large unwrappings. By the conditional correlation lemma, we know we can calculate $|\tilde{C}_{x_1|y}r|$ by constructing an unwrapped tree with observations r and computing the conditional mean at the root node. Since the unwrapped tree is singly-connected, we can calculate the mean at the root node exactly by running belief propagation. We start by sending messages from the leaf nodes to the layer above and continue sending messages upwards until we reach the root node.

We use M_l to denote the maximum absolute value of means of all messages sent upward by layer l. By equation 18 the value μ_0 calculated by all nodes at layer $l-1$ are a weighted average of means from the previous layer and the means from the observations at layer $l-1$. Since r is zero for all layers but the bottom one, the means of the messages sent by observations at the $l-1$ layer are zero. Thus:

$$|\mu_0| \leq M_l \tag{43}$$

If we rewrite equation 16 taking advantage of the fact that x_i, x_j are scalars we have:

$$\mu_{ij} = \frac{-b(a+P_0)^{-1}P_0\mu_0}{c - b^2(a+P_0)^{-1}} \tag{44}$$

Multiplying top and bottom by $\frac{(a+P_0)}{P_0}$:

$$\mu_{ij} = \frac{-b}{(ac-b^2)/P_0 + c}\mu_0 \tag{45}$$

Similarly, we can rewrite equation 15 taking advantage of x_i, x_j being scalar to give:

$$P_{ij} = \frac{ca - b^2 + cP_0}{a + P_0} \tag{46}$$

Note that for diagonally dominant matrices, P_{ij} is non-negative if P_0 is non-negative. Note also that if $V_{ii}(1,1) > 0$for all the observation potentials, the observations will send a positive precision to the unobserved nodes. Thus all precisions will be non-negative.

Since V_{ij} is diagonally dominant, and P_0 is non-negative, both terms in the denominator of equation 45 are non-negative so:

$$|\mu_{ij}| \leq \frac{|b|}{|c|}|\mu_0| \tag{47}$$

We now denote by $\beta = \max_{i,j} |V_{ij}(1,2)/V_{ij}(2,2)|$. Since all V_{ij} are diagonally dominant, $\beta < 1$. Combining equations 43,47 gives:

$$M_{l-1} \leq \beta M_l \tag{48}$$

So for any ϵ if we choose $t_\epsilon > \frac{\log \epsilon}{\log \beta}$ then $|\tilde{C}_{x_1|y}r| < \epsilon \max|r(i)|$. □.

The conditional correlation lemma can also be used to give bounds on the loopy variances. For example:

Corollary: If V_{ij} are diagonally dominant and the off-diagonal elements are negative then the loopy beliefs are overconfident $\tilde{\sigma}^2(1) < \sigma^2(1)$.

Proof: By the conditional correlation lemma we can calculate $\tilde{C}_{x_1|y}r_2$ by setting the observations to be one at all copies of the root node and zero elsewhere. From equation 45 it is clear that when $b < 0$ and the observations are zero or one, the mean messages are weighted averages of positive values hence the mean at the root will be positive. □

As we discussed in section 1 the decomposition of a Gaussian MRF into V_{ij}, V_{ii} is not unique. The following lemma shows that even when we have a loopy graph where V_{ij} are not diagonally dominant, belief propagation will still converge when there exists a reparameterization using diagonally dominant matrices.

Reparameterization Lemma: If the unwrapped tree for any iteration of loopy propagation can be parameterized so that $\tilde{V}_{ij}$ are diagonally dominant and $\tilde{V}_{ii}(1,1) > 0$ then (1) belief propagation converges (2) the belief propagation means are exact and (3) the belief propagation variances are equal to the correct variances minus the summed conditional correlations between $\tilde{x}_1$ and all $\tilde{x}_j$ that are replicas of x_1.

Proof: This follows from the proof of claim 4. Since the unwrapped tree is singly connected, belief propagation using any parameterization is exact. So we can calculate $\tilde{C}_{x_1|y}r$ by running belief propagation on the reparameterized tree in which all the matrices are diagonally dominant. □

We emphasize that claim 4 only gives sufficient conditions for convergence. It is easy to construct networks in which V_{ij} are not all diagonally dominant but loopy belief propagation still converges. In section we show an example.

3 Fixed points of loopy propagation

Each iteration of belief propagation can be thought of as an operator F that inputs a list of messages $m^{(t)}$ and outputs a list of messages $m^{(t+1)} = Fm^{(t)}$. Thus belief propagation can be thought of as an iterative way of finding a solution to the fixed point equations $Fm = m$ with an initial guess m_0 in which all messages are constant functions.

Note that this is not the only way of finding fixed-points. McEliece et al. [17] have shown a simple example for which F contains multiple fixed points and belief propagation finds only one. They also showed an example where a fixed-point exists but the iterations $m \leftarrow Fm$ do not converge. Murphy et al. (1999) describe an alternative method for finding fixed-points of F.

In this section we ask, suppose a fixed-point $m^* = Fm^*$ has been found by some method, how are the beliefs calculated based on these messages related to the correct beliefs?

Claim 5: For a Gaussian graphical model of arbitrary topology, if m^* is a fixed-point of the message-passing dynamics then the means based on that fixed-point are exact.

The proof is based on the following lemma:

Periodic beliefs lemma: If m^* is a fixed-point of the message-passing dynamics

in a graphical model G then one can construct a modified unwrapped tree T of arbitrarily large depth such that: (1) all non-leaf nodes in T have the same statistical relationship with their neighbors as the corresponding nodes in G and (2) all nodes in T will have the same belief as the beliefs in G derived from m^*.

Proof: The proof is by construction. We first construct an unwrapped tree T of the desired depth. We then modify the potentials and the observations in the leaf nodes in the following manner. For each leaf node $\tilde{x}_i$, find the $k-1$ nodes in G that neighbor $x_{i'}$ (where $\tilde{x}_i$ is a replica of $x_{i'}$) excluding the parent of $\tilde{x}_i$. Calculate the product of the $k-1$ messages that these neighbors send to the corresponding node in G under the fixed-point messages m^* and the message that $y_{i'}$ sends to $x_{i'}$. Set $\tilde{y}_i$ and $\Psi(\tilde{y}_i, \tilde{x}_i)$ such that the message $\tilde{y}_i$ sends to $\tilde{x}_i$ is equal to this product.

By this construction, all leaf nodes in T will send their neighbors a message from m^*. Since all non-leaf nodes in T have the same statistical relationship to their neighbors as the corresponding nodes in G, the local message passing updates in T are identical to those in G. Thus all messages in T will be replicas of messages in m^*. □

Proof of Claim 5: Using this lemma we can prove claim 5. Let $\tilde{\mu}$ be the conditional mean in the modified unwrapped tree then, by the periodic beliefs lemma:

$$\tilde{\mu} = O\mu_0 \tag{49}$$

where $\mu_0(i)$ is the posterior mean at node i under m^*.

We also know that $\tilde{\mu}$ is a solution to:

$$\tilde{V}_{xx}\tilde{\mu} = -\tilde{V}_{xy}\tilde{y} \tag{50}$$

where $\tilde{V}_{xx}, \tilde{V}_{xy}, \tilde{y}$ refers to quantities in the modified unwrapped tree. So:

$$\tilde{V}_{xx}O\mu_0 = -\tilde{V}_{xy}\tilde{y} \tag{51}$$

We use the notation $[A]_m$ to indicate taking the m first rows of a matrix A. Note that for any two matrices $[AB]_m = [A]_m B$. Taking the first m rows of equation 51 gives:

$$\left[\tilde{V}_{xx}O\right]_m \mu_0 = -\left[\tilde{V}_{xy}\tilde{y}\right]_m \tag{52}$$

As in the previous proofs, the key idea is to relate the inverse covariance matrix of the modified unwrapped tree to that of the original loopy graph. Since all non-leaf nodes in the modified unwrapped tree have the same neighborhood relationships with their neighbors as the corresponding nodes in the loopy graph we have, for any $m < N - L$:

$$\left[\tilde{V}_{xx}O\right]_m = [OV_{xx}]_m \tag{53}$$

and:

$$\left[\tilde{V}_{xy}\tilde{y}\right]_m = [OV_{xy}y]_m \tag{54}$$

Substituting these relationships into equation 52 gives:

$$[O]_m V_{xx}\mu_0 = -[O]_m V_{xy}y \tag{55}$$

This equation holds for any $m < N - L$. Since we can unwrap the tree to arbitrarily large size we can choose m such that $[O]_m$ has n independent rows (this happens once all nodes in the loopy graph appear at least once in the modified unwrapped tree). Thus:

$$V_{xx}\mu_0 = -V_{xy}y \tag{56}$$

hence the means derived from the fixed-point messages are exact. □

4 Extensions

4.1 Non-Gaussian variables

In Sect. 1 we described the "max-product" belief propagation algorithm that finds the MAP estimate for each node [20, 24] of a network without loops. As with max-product, iterating this algorithm is a method of finding a fixed-point of the message passing dynamics. How does the assignment derived from this fixed-point compare the MAP assignment?

Claim 6: For a graphical model of arbitrary topology with continuous potential functions, if m^* is a fixed-point of the max-product message-passing dynamics then the assignment based on that fixed-point is a local maximum of the posterior probability.

Proof: Since the posterior probability factorizes into a product of pairwise potentials, the log posterior will have the form,

$$\log P(x|y) = \sum_{ij} J_{ij}(x_i, x_j) + J_{ii}(x_i, y_i) \tag{57}$$

Assuming the cliiue potential functions are differentiable and finite, the MAP solution, u, will satisfy

$$\frac{\partial}{\partial x_i} \log P(x|y)|_{x=u} = 0 \tag{58}$$

We will write this as:

$$Vu = 0 \tag{59}$$

where V is a *nonlinear* operator.

As in the previous section, we can use the periodic belief lemma to construct a modified unwrapped tree of arbitrary size based on m^*. If we denote by $\tilde{V}$ the nonlinear set of equations that the solution to the modified unwrapped problem must satisfy we have:

$$\tilde{V}\tilde{u} = 0 \tag{60}$$

Because of the periodic belief lemma:

$$\tilde{u} = Ou_0 \tag{61}$$

Similarly, as in the previous section, all the non-leaf nodes will have the same statistical relationship with their neighbors as do the corresponding nodes in the loopy network, so:

$$\left[\tilde{V}O\right]_m = [OV]_m \tag{62}$$

where the left and right hand sides are nonlinear operators.

Substituting Eqs. 61 and 62 into Eq. 60 gives:

$$Vu_0 = 0 \tag{63}$$

A similar substitution can be made with the second derivative equations to show that the Hessian at u_0 is positive definite. Thus the assignment based on m^* is at least a local maximum of the posterior. □

Claim 6 can be generalized to discrete nodes with a more general definition of local maximum.

Definition: A discrete assignment x is a generalized local maximum of $prob(x)$ with respect to a set of allowed moves $\{\Delta_i\}$ if for all i $prob(x) > prob(x + \Delta_i)$

Claim 7: For a graphical model of arbitrary topology with discrete nodes, if m^* is a fixed-point of the max-product message-passing dynamics then x^*, the assignment based on that fixed-point, is a generalized local maximum of the posterior probability for any set of moves $\{\Delta_i\}$ that only allow changing singly-connected subtrees of x^*.

Proof: Define

$$Vu = \arg\max_{\Delta i} Prob(x = u + \Delta_i | y) \tag{64}$$

So at a local maximum $Vu = 0$. The rest of the proof is analogous to the proof of claim 6.

4.2 Vector valued nodes

Most of the results we have derived so far hold for vector-valued nodes as well but the indexing notation is rather more cumbersome. We use a stacking convention, in which we define the vector x by:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \end{pmatrix} \tag{65}$$

Thus supposing x_1 is a vector of length 2 then $x(1)$ is the first component of x_1 and $x(2)$ is the second component of x_1 (not x_2). We define y in a similar fashion.

Using this stacking notation the equations for exact inference in Gaussians remain unchanged, but we need to be careful in reading out the posterior means and covariances from the stacked vectors. Thus we can still complete the square in stacked notation to obtain:

$$V_{xx}\mu = -V_{xy}y \tag{66}$$

and $C_{x|y} = V_{xx}^{-1}$. Assuming x_1 is of length 2, μ_1 the posterior mean of x_1 is given by:

$$\mu_1 = \begin{pmatrix} \mu(1) \\ \mu(2) \end{pmatrix} \tag{67}$$

and the posterior covariance matrix Σ_1 is given by:

$$\Sigma_1 = \begin{pmatrix} C_{x|y}(1,1) & C_{x|y}(1,2) \\ C_{x|y}(2,1) & C_{x|y}(2,2) \end{pmatrix} \tag{68}$$

We use the same stacked notation for $\tilde{x}$ and define the matrix O such that $O(i,j) = 1$ if $\tilde{x}(i)$ is a replica of $x(j)$ and zero otherwise. Using this notation, the relationships between unwrapped and loopy quantities (e.g. $\left[\tilde{V}_{xx}O\right]_m = [OV_{xx}]_m$) still hold. Thus all the analysis done in the previous sections holds — the only difference are the semantics of quantities such as $\mu(1)$, which need to be understood as a scalar component of a (possibly) larger vector μ_1. For explicitness, we restate the five claims for vector valued nodes.

For any i, j less than or equal to the number of components in x_1 we have:

Claim 1a:

$$\tilde{\mu}(i) = \mu(i) + \tilde{C}_{x_i|y} r \tag{69}$$

where r is a vector that is zero everywhere but the last L components (corresponding to the leaf nodes).

Claim 2a:

$$\tilde{C}_{x|y}(i,j) = C_{x|y}(i,j) + \tilde{C}_{x_j|y} r_1 - \tilde{C}_{x_j|y} r_2 \tag{70}$$

where r_1 is a vector that is zero everywhere but the last L components (corresponding to the leaf nodes) and r_2 is equal to 1 for all components corresponding to non-root nodes in the unwrapped tree that reference $x(i)$. All other components of r_2 are zero.

Claim 3a: If the conditional correlation between all components of the root node and the leaf nodes decreases rapidly enough then (1) belief propagation converges (2) the belief propagation means are exact and (3) the i, j component of the belief propagation covariance matrices is equal to the i, j component of the true covariance matrices minus the summed conditional correlations between $\tilde{x}(j)$ and all nonroot $\tilde{x}(k)$ that are replicas of $x(i)$.

Claim 5a: For a (possibly vector-valued) Gaussian graphical model of arbitrary topology, if m^* is a fixed-point of the message-passing dynamics, then the means based on that fixed-point are exact.

Claim 6a: For a (possibly vector-valued) graphical model of arbitrary topology with continuous potential functions, if m^* is a fixed-point of the max-product message-passing dynamics, then the assignment based on that fixed-point is a local maximum of the posterior probability.

We emphasize that these claims do not need to be reproven — all the equations used in proving the scalar-valued case still hold only the semantics we place on the individual components are different.

We end this analysis with two simple corollaries:

Corollary 1: Let m^* be a fixed-point of Pearl's belief propagation algorithm on a Gaussian Bayesian network of arbitrary topology and arbitrary clique size. Then the means based on m^* are exact.

Corollary 2: Let m^* be a fixed-point of Pearl's belief revision (max-product) algorithm on a Bayesian network with continuous joint probability, arbitrary topology and arbitrary clique size. The assignment based on m^* is at least a local maximum of the posterior probability.

These corollaries follow from claims 5a and 6a along with the equivalence between

Pearl's propagation rules and the propagation rules for pairwise undirected graphical models analyzed here [24]. Note that even if the Bayesian network contained only scalar nodes, the conversion to pairwise cliques may necessitate using vector-valued nodes.

5 Simulations

To illustrate the analysis, we ran belief propagation on a 25×25 2D grid. The joint probability was:

$$P(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\sum_{ij} w_{ij}(x_i - x_j)^2 - \sum_i w_{ii}(x_i - y_i)^2) \tag{71}$$

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 10^{-6} or 1 for all i with probability of 1 set to 0.2. When $w_{ii} = 1$, we set the observations y_i to be samples from the surface $z(x, y) = x + y$. When $w_{ii} = 10^{-6}$ we set y_i=0. This problem corresponds to an approximation problem from sparse data where only 20% of the points are visible and there is a weak prior on the unobserved nodes pulling them towards zero.

We found the exact posterior by solving Eq. 12. We also ran loopy belief propagation and found that when it converged, the loopy means were identical to the true means up to machine precision. Also, as predicted by the theory, the loopy variances were too small — the loopy estimate was overconfident.

How is this predicted from our analysis? This illustrates the reparameterization lemma in section 2.5. The parameterization we used in the loopy network was $V_{ij} = \begin{pmatrix} w_{ij} & -w_{ij} \\ -w_{ij} & w_{ij} \end{pmatrix}$, $V_{ii} = \begin{pmatrix} w_{ii} & -w_{ii} \\ -w_{ii} & w_{ii} \end{pmatrix}$. Thus V_{ij} are not diagonally dominant. However, we can also use a different parameterization in which we shift the diagonal component from V_{ii} to V_{ij}: $\tilde{V}_{ij} = \begin{pmatrix} w_{ij} + \frac{w_{ii} - \epsilon}{deg(x_i)} & -w_{ij} \\ -w_{ij} & w_{ij} + \frac{w_{jj} - \epsilon}{deg(x_j)} \end{pmatrix}$, $\tilde{V}_{ii} = \begin{pmatrix} \epsilon & -w_{ii} \\ -w_{ii} & w_{ii} \end{pmatrix}$. In this parameterization, all $\tilde{V}_{ij}$ are diagonally dominant. Thus claim 4 guarantees that belief propagation will converge and the variances will be overconfident.

We also ran belief propagation on this problem with $w_{ii} = 0$ for the unobserved nodes. Note that under this case we cannot reparameterize the unwrapped tree so that all V_{ij} are diagonally dominant so claim 4 does not hold. We found that belief propagation still converged with a similar convergence rate and that the means were exact. This illustrates the fact that claim 4 only gives sufficient (but not necessary) conditions for the conditional correlation to decrease rapidly enough. Claim 5 guarantees that the means will be exact.

In many applications, the solution of equation 12 by matrix inversion is intractable and iterative methods are used. Figure 4 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 [21]. We used an over-relaxation constant of 1.9. Note that after five iterations loopy propagation has mean squared error of order 10^{-1} while SOR requires many more. As expected by the fast convergence, the approximation error in the variances was quite small. The median

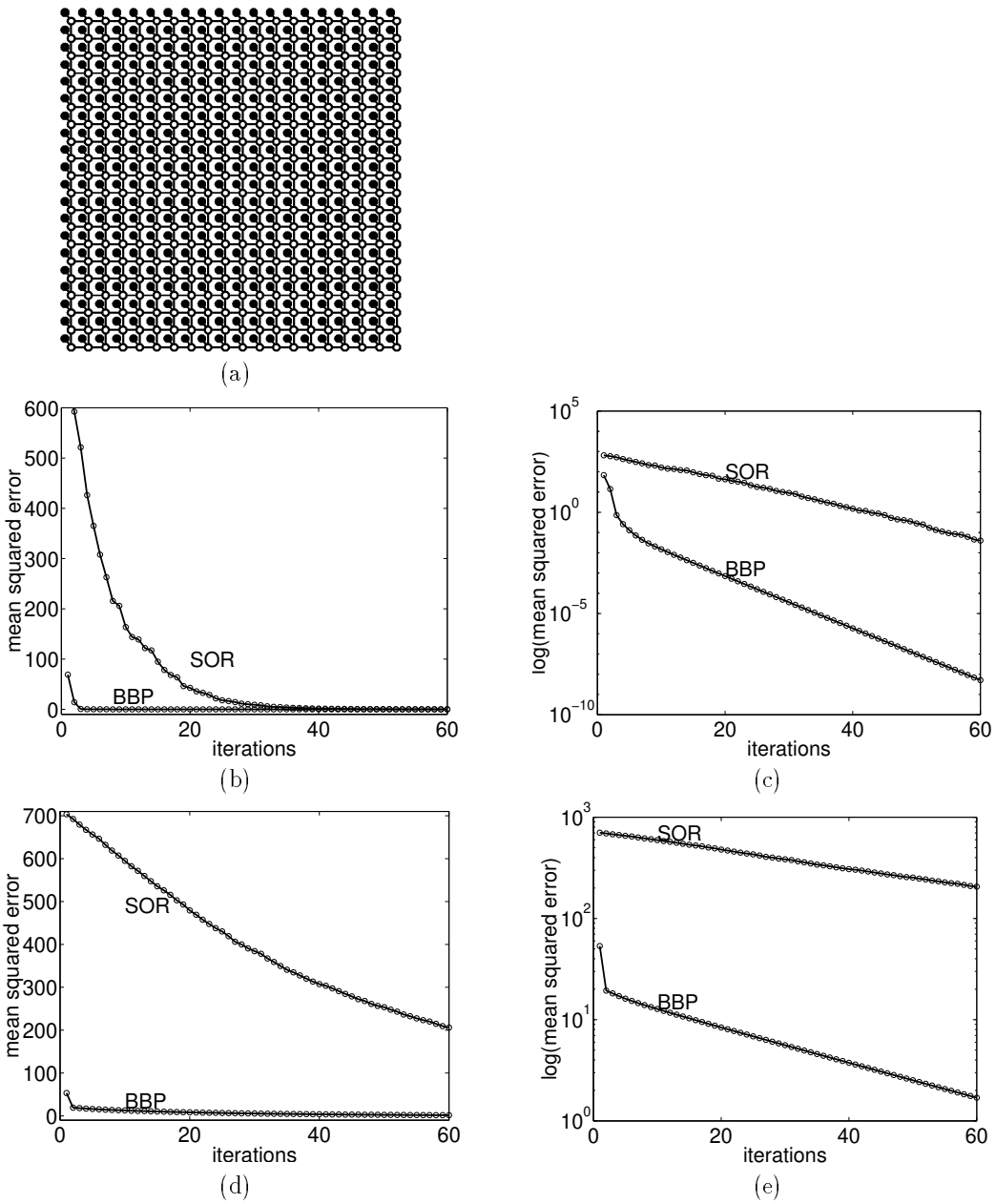

Figure 4: (a) 25×25 graphical model for simulation. The unobserved nodes (unfilled) were connected to their four nearest neighbors and to an observation node (filled). (b) The error of the estimates of loopy propagation and successive overrelaxation (SOR) as a function of iteration. Note that belief propagation converges much faster than SOR. (c) the same plot as in (b) but with log scaling on the y axis (d-e) similar plots when the connections between unobserved nodes were increased by four orders of magnitude. Convergence of both BBP and SOR is much slower in this case but BBP is still faster.

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.

The analysis in section 2.5 suggests that the convergence rate is related to the ratio of diagonal to off-diagonal elements in $\tilde{V}_{ij}$. To illustrate this, We redid the simulations with the same network but set $w_{ij} = 10.0$ between neighboring nodes. Indeed convergence is much slower in this case (figure 4d,e). As expected from the slower convergence, the error in the variance estimates is larger here with the median error 0.299.

The slow convergence of SOR on problems such as these lead to the development of multi-resolution models in which the MRF is approximated by a tree [15, 5] and an algorithm equivalent to belief propagation is then run on the tree. Although the multi-resolution models are much more efficient for inference, the tree structure often introduces block artifacts in the estimate. Our results suggest that one can simply run belief propagation on the original MRF and get the exact posterior means. It would be interesting to see whether one can use the convergence rate of the beliefs to improve the variance estimate at every node.

6 Discussion

Independent of our results, two groups have recently analyzed special cases of Gaussian graphical models. Frey [8] analyzed the graphical model corresponding to factor analysis and gave conditions for the existence of a stable fixed-point. Rusmevichientong and Van Roy [19] analyzed a graphical model with the topology of turbo decoding but a Gaussian joint density. They showed that for this specific case, belief propagation converges and the means are exact.

Our main interest in analyzing the Gaussian case was to understand the performance of belief propagation in networks with multiple loops. Although there are many special properties of Gaussians, we are struck by the similarity of the analytical results reported here for multi-loop Gaussians and the analytical results for single loops and general distributions reported in [24]. The most salient similarities are:

- In single loop networks with binary nodes, the mode at each node is guaranteed to be correct but the confidence in the mode may be incorrect. In Gaussian networks with multiple loops the mean at each node is guaranteed to be correct but the confidence around that mean will in general be incorrect.

- In single loop networks fast convergence is correlated with good approximation of the beliefs. This is also true for Gaussian networks with multiple loops.

- In single loop networks the convergence rate and the approximation error were determined by a ratio of eigenvalues λ_1/λ_2. This ratio determines the extent of the statistical dependencies between the root and the leaf nodes in the unwrapped network for a single loop. In Gaussian networks the convergence rate and the approximation error are determined by the off-diagonal terms of $\tilde{C}_{x|y}$. These terms quantify the extent of conditional

dependencies between the root nodes and the leaf nodes of the unwrapped network.

These similarities are even more intriguing when one considers how different Gaussians graphical models are from discrete models with arbitrary potentials and a single loop. In Gaussians the conditional mean is equal to the conditional mode and there is only one maximum in the posterior probability, while the single loop discrete models may have multiple maxima, none of which will be equal to the mean. Furthermore, in terms of approximate inference the two classes behave quite differently. For example, mean field approximations give the exact means for Gaussian MRFs while they work poorly in discrete networks with a single loop in which the connectivity is sparse [22]. The resemblance of the results for Gaussian graphical models and for single loops leads us to believe that similar results may hold for a larger class of networks.

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

Acknowledgments

We thank A. Ng, K. Murphy, P. Pakzad, B. Frey and M.I. Jordan for comments on previous versions of this manuscript. YW is supported by MURI-ARO-DAAH04-96-1-0341

References

[1] S. M. Aji and R.J. McEliece. The generalized distributive law. IEEE Transactions on Information Theory, 1999. to appear.

[2] 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.

[3] 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.

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

[5] M. M. Daniel and A. S. Willsky. The modeling and estimation of statistically self-similar processes in a multiresolution framework. *IEEE Trans. Info. Theory*, 45(3):955–970, April 1999.

[6] 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.

[7] 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.

[8] B.J. Frey. Turbo factor analysis. In *Adv. Neural Information Processing Systems 12*. 1999. submitted.

[9] Brendan J. Frey. *Graphical Models for Pattern Classification, Data Compression and Channel Coding*. MIT Press, 1998.

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

[11] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, November 1984.

[12] F.V. Jensen. *An Introduction to Bayesian Networks*. Springer, 1996.

[13] 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.

[14] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[15] Mark R. Luettgen, W. Clem Karl, and Allan S. Willsky. Efficient multiscale regularization with application to the computation of optical flow. *IEEE Transactions on image processing*, 3(1):41–64, 1994.

[16] 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.

[17] 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.

[18] 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.

[19] Rusmevichientong P. and Van Roy B. An analysis of Turbo decoding with Gaussian densities. In *Adv. Neural Information Processing Systems 12*. 1999. submitted.

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

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

[22] Y. Weiss. Interpreting images by propagating Bayesian beliefs. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, 1996.

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

[24] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, to appear, 1999.

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