Loop Corrected Belief Propagation

Joris Mooij, Bastian Wemmenhove, Bert Kappen
Department of Biophysics
Radboud University Nijmegen
6525 EZ Nijmegen, The Netherlands
{j.mooij|b.wemmenhove|b.kappen}@science.ru.nl

Tommaso Rizzo
Laboratoire de Physique Théorique
de l'Ecole Normale Supérieure
24 rue Lhomond, 75231 Paris, France
rizzo@lpt.ens.fr

Abstract

We propose a method for improving Belief Propagation (BP) that takes into account the influence of loops in the graphical model. The method is a variation on and generalization of the method recently introduced by Montanari and Rizzo [1]. It consists of two steps: (i) standard BP is used to calculate *cavity distributions* for each variable (i.e. probability distributions on the Markov blanket of a variable for a modified graphical model, in which the factors involving that variable have been removed); (ii) all cavity distributions are combined by a message-passing algorithm to obtain consistent single node marginals. The method is exact if the graphical model contains a single loop. The complexity of the method is exponential in the size of the Markov blankets. The results are very accurate in general: the error is often several orders of magnitude smaller than that of standard BP, as illustrated by numerical experiments.

1 INTRODUCTION

Belief Propagation (BP), also known as the Sum-Product Algorithm and as Loopy Belief Propagation, is a popular algorithm for approximate inference on graphical models. It often yields surprisingly accurate results, using little computation time. It has strong ties with the Bethe approximation [2], which was developed in statistical physics [3]. Belief Propagation is the simplest case in a family of more sophisticated algorithms such as Generalized Belief Propagation (GBP) [4] (which can be used e.g. for the Cluster Variation Method (CVM) [5, 6]) and Expectation Propagation (EP) [7].

It is well-known that Belief Propagation yields exact results if the graphical model is a tree. However, if the graphical model contains loops (cycles), the approximate marginals calculated by BP can have large errors. Increasing the cluster size of the approximation (e.g. by using CVM with larger clusters) does not necessarily solve this problem if long, influential loops cannot be completely included in one cluster. Using TreeEP [8] one can correct for the presence of loops to a certain extent, namely for those loops that consist of part of the base tree and one additional factor. The method we propose here effectively takes into account all the loops in the factor graph, in many cases yielding more accurate approximate marginals as a result.

In the statistical physics community different methods for calculating loop corrections to the Bethe approximation have been proposed recently [1, 9, 10]. The work we present here is a variation on the theme introduced in [1]. The alternative that we propose here offers two advantages compared to the original method proposed in [1]: (i) it is more robust and improves upon BP also for relatively strong interactions; (ii) it is directly applicable to arbitrary factor graphs, whereas the original method has only been formulated for binary variables with pairwise factors.

This article is organised as follows. First we explain the theory behind the proposed method, discussing differences with the original approach in [1] along the way. Then we report numerical experiments regarding the quality of the approximation and the computation time, comparing with other approximate inference methods. Finally, we discuss the results and state conclusions.

2 THEORY

2.1 GRAPHICAL MODEL CLASS, NOTATIONS

Let $\mathcal{V} := \{1, \dots, N\}$ be an index set for N random variables $\{x_i\}_{i \in \mathcal{V}}$, where variable x_i takes values in a discrete domain $\mathcal{X}_i$. We will use a multi-index notation, i.e. for any subset $I \subseteq \mathcal{V}$, we write $x_I := (x_{i_1}, x_{i_2}, \dots, x_{i_m})$ if $I = \{i_1, i_2, \dots, i_m\}$ and $i_1 < i_2 < \dots i_m$. We consider probability distributions over $x = (x_1, \dots, x_N)$ that can be written as a product of small factors ψ_I:

$$P(x_1, \dots, x_N) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I). \tag{1}$$

The factors (or "interactions") ψ_I are indexed by subsets of $\mathcal{V}$, i.e. $I \in \mathcal{F} \subseteq \mathcal{P}(\mathcal{V})$. Each factor is a non-negative function $\psi_I : \prod_{i \in I} \mathcal{X}_i \to [0, \infty)$. This class of probability distributions includes Markov Random Fields as well as Bayesian Networks. In general, the normalizing constant Z is not known and exact computation of Z is infeasible. One can visualize a probability distribution of the form (1) with a *factor graph* (c.f. Figure 1(a)), a bipartite graph having *variable nodes* $i \in \mathcal{V}$ and *factor nodes* $I \in \mathcal{F}$, with an edge between i and I if and only if $i \in I$.

In the following, we will use uppercase letters for indices of factors ($I, J, K, \dots \in \mathcal{F}$) and lowercase letters for indices of variables ($i, j, k, \dots \in \mathcal{V}$). For simplicity we assume that no pair of variables is contained in more than one factor, i.e. we assume that no loops of length 4 are present in the factor graph. We slightly abuse notation by writing $\setminus I$ instead of $\mathcal{V} \setminus I$ for $I \subseteq \mathcal{V}$ and $\setminus i$ instead of $\mathcal{V} \setminus \{i\}$ for $i \in \mathcal{V}$.

2.2 LCBP: A BRIEF OVERVIEW

The main idea of what is known in the statistical physics community as the "cavity method" is to consider modified graphical models in which a single variable is removed, together with all factors in which that variable appears, thus forming a "cavity" (c.f. Figure 1). The removed variable is called the *cavity variable.* The method proposed in [1] (and our method, which is a variation and generalization thereof) approximates for each variable its corresponding *cavity distribution*, i.e. the marginal probability distribution of the cavity network on the neighbourhood (Markov blanket) of the cavity variable. Subsequently, the removed factors are multiplied back in, and we demand consistency of single node marginals. This results in partial cancellation of errors in the approximated cavity distributions, improving the accuracy of the final result. The Bethe approximation is obtained as the special case in which the cavity distributions are assumed to factorize completely. We will now explain the procedure in more detail.

2.3 CAVITIES

Let $i \in \mathcal{V}$. We denote by $\partial i := \{j \in \mathcal{V} : i, j \in I \text{ for some } I \in \mathcal{F}\}$ the set of neighbouring variables of i, also called the *Markov blanket* of i. We define $\Delta i := \partial i \cup \{i\}$. We modify the original graphical model (1) by removing variable x_i and all the factors in which it appears (c.f. Figure 1); the probability distribution corresponding to the resulting *cavity network* is thus by definition:

$$\frac{1}{Z_{\setminus i}} \prod_{\substack{I \in \mathcal{F} \\ i \notin I}} \psi_I(x_I). \tag{2}$$

Note that the normalization constant $Z_{\setminus i}$ differs from the normalization constant Z of the original network (1). We will call the marginal distribution of (2) on ∂i (the Markov blanket of i) the *cavity distribution* $P^{\setminus i}$ *of* i:

$$P^{\setminus i}(x_{\partial i}) := \frac{1}{Z_{\setminus i}} \sum_{x_{\setminus \Delta i}} \prod_{\substack{I \in \mathcal{F} \\ i \notin I}} \psi_I(x_I). \tag{3}$$

Writing Ψ_i for the product of the removed factors:

$$\Psi_i(x_{\Delta i}) := \prod_{\substack{I \in \mathcal{F} \\ i \in I}} \psi_I(x_I), \tag{4}$$

the following identity is immediate:

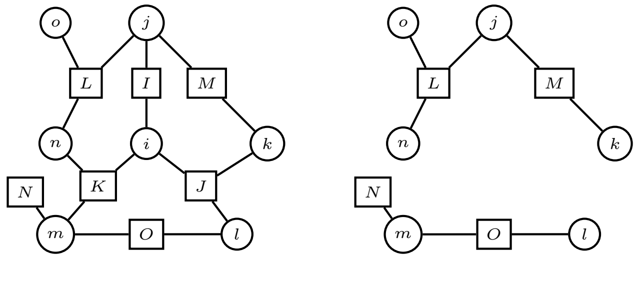
$$P(x_{\Delta i}) \propto P^{\setminus i}(x_{\partial i}) \Psi_i(x_{\Delta i}), \tag{5}$$

i.e. the marginal distribution on Δi of the *original* probability distribution (1) is proportional to the product of the cavity distribution of i and the product of the factors involving x_i.[1] The cavity distribution summarizes the rest of the network; it can be seen as an "effective interaction" on $x_{\partial i}$. In particular, it summarizes information about loops in which variable i is contained. For example, in Figure 1, the cavity distribution $P^{\setminus i}(x_{\partial i})$ contains the interaction ψ_O between x_m and x_l, which is part of the loop $iKmOlJi$ in the original factor graph.

2.4 CONSISTENCY OF SINGLE NODE MARGINALS

Consider two variables $i, j \in \mathcal{V}$ with $i \in \partial j$; let I be the common factor involving both x_i and x_j. The joint

[1] Note that equation (5) is *not* one of the DLR equations [11]. The most similar DLR equation would be $P(x_i) = \sum_{x_{\partial i}} P(x_i \,|\, x_{\partial i}) P(x_{\partial i})$, whereas (5) implies $P(x_i) \propto \sum_{x_{\partial i}} \Psi_i(x_{\Delta i}) P^{\setminus i}(x_{\partial i})$. Although the equations may appear identical at first sight, they are not. Consider e.g. a star-shaped model with a central variable i coupled to its neighbours by pairwise factors. In that case, $P^{\setminus i}(x_{\partial i})$ is completely factorized, whereas $P(x_{\partial i})$ is not.



(a) Original graph (b) Cavity graph of i

Figure 1: (a) Original factor graph; (b) cavity graph remaining after variable node i and the factor nodes that contain i (i.e. I,J,K) have been removed. The neighbours of i are $\partial i = \{j,k,l,m,n\}$ and $\Delta i = \{i,j,k,l,m,n\}$. The cavity distribution $P^{\setminus i}$ is defined as the marginal on $x_{\partial i}$ of the probability distribution corresponding to (b).

marginal on x_i and x_j *in the absence of the factor I* is given by

$$P^{\setminus I}(x_i,x_j) \propto \sum_{x_{\setminus\{i,j\}}} \prod_{\substack{J\in\mathcal{F}\\ J\neq I}} \psi_J(x_J). \tag{6}$$

We can calculate this joint marginal from the cavity distribution of i:

$$P^{\setminus I}(x_i,x_j) \propto \sum_{x_{\partial i\setminus j}} P^{\setminus i}(x_{\partial i})\Psi_i^{\setminus I}(x_{\Delta i}) \tag{7}$$

where we defined:

$$\Psi_i^{\setminus I}(x_{\Delta i}) := \prod_{\substack{J\in\mathcal{F}\\ i\in J, J\neq I}} \psi_J(x_J) = \frac{\Psi_i}{\psi_I} \qquad \text{for } i\in I, I\in\mathcal{F}.$$

Alternatively, we can calculate (6) from the cavity distribution of j (interchanging i and j):

$$P^{\setminus I}(x_i,x_j) \propto \sum_{x_{\partial j\setminus i}} P^{\setminus j}(x_{\partial j})\Psi_j^{\setminus I}(x_{\Delta j}). \tag{8}$$

The results are obviously identical if the cavity distributions $P^{\setminus i}$ and $P^{\setminus j}$ are exact.

In practice, the exact cavity distributions are unavailable and we can only obtain approximations $Q_0^{\setminus k} \approx P^{\setminus k}$. Replacing $\{P^{\setminus k}\}_{k\in\mathcal{V}}$ by their approximations $\{Q_0^{\setminus k}\}_{k\in\mathcal{V}}$ in equations (7) and (8) will yield inconsistent results; the main idea of the method proposed in [1] is to deform the approximate cavity distributions $\{Q_0^{\setminus k}\}_{k\in\mathcal{V}}$ in such a way that the single node marginals of x_i and x_j in equations (7) and (8) become consistent.[2] In [1], the single node marginals of the approximate cavity distributions are varied whereas the higher order cumulants are kept fixed.[3] Instead, we propose here to deform the approximate cavity distributions $Q_0^{\setminus i}$ in the following way:

$$Q^{\setminus i}(x_{\partial i}) \propto Q_0^{\setminus i}(x_{\partial i}) \prod_{j\in\partial i} \phi_j^{\setminus i}(x_j). \tag{9}$$

Thus we change the single variable interactions by multiplying with single node factors but keep higher order interactions fixed. The single node factors $\phi_j^{\setminus i}(x_j)$ are chosen such that the single node marginals of x_i and x_j are consistent in the absence of factor I, i.e. such that

$$\sum_{x_{\Delta i\setminus i}} Q^{\setminus i}(x_{\partial i})\Psi_i^{\setminus I}(x_{\Delta i}) \propto \sum_{x_{\Delta j\setminus i}} Q^{\setminus j}(x_{\partial j})\Psi_j^{\setminus I}(x_{\Delta j}). \tag{10}$$

This should hold for all pairs of neighbouring variables $i,j\in\mathcal{V}$ with $i\in\partial j$. In this way, first order errors in the initial approximate cavity distributions $Q_0^{\setminus k}$ are cancelled out.

To calculate the values for the corrections $\phi_j^{\setminus i}(x_j)$, we use Algorithm 1, which is a simple fixed-point algorithm based on equations (10). After convergence, we calculate single node marginals $q_i(x_i) \approx P(x_i)$ from the final deformed approximate cavity distributions $Q_\infty^{\setminus i}$ using:

$$q_i(x_i) \propto \sum_{x_{\partial i}} Q_\infty^{\setminus i}(x_{\partial i})\Psi_i(x_{\Delta i}).$$

In our experiments, Algorithm 1 always converged to a reproducible fixed point, even without damping. Note that if we would start with the exact cavity distributions, i.e. $Q_0^{\setminus i} = P^{\setminus i}$ for all i, the algorithm would terminate immediately because the single node marginals would already be consistent. Obviously, one can use other update schemes than the parallel one given in Algorithm 1; in our experiments, we have used a sequential update scheme.

[2] Instead of demanding consistency of single node marginals x_i and x_j in the *absence* of the factor I connecting x_i with x_j, one could alternatively demand consistency of the single node marginals in the presence of all factors, i.e. demanding that $\sum_{x_{\partial i\setminus j}} P^{\setminus i}(x_{\partial i})\Psi_i(x_{\Delta i}) \propto \sum_{x_{\partial j\setminus i}} P^{\setminus j}(x_{\partial j})\Psi_j(x_{\Delta j})$ for all $i\in\mathcal{V}, j\in\partial i$. This might appear more natural, but it turns out that the resulting method is inferior to the one presented here if factors involving more than two variables are present (see also section 4).

[3] Cumulants are called "connected correllations" in [1] and are defined as certain polynomial combinations of moments $\sum_{x_{\partial i}} P^{\setminus i}(x_{\partial i})\prod_{j\in\mathcal{A}} x_j$ with $\mathcal{A}\subseteq\partial i$, where all variables are assumed to be ±1-valued.

2.5 COMPUTING $Q_0^{\setminus i}$

We have discussed in the previous subsection how to deform the initial approximate cavity distributions $Q_0^{\setminus i}$ to make them consistent; we now discuss how to obtain the $Q_0^{\setminus i}$ in the first place.

In [1] it is suggested to initialize the second-order cumulants of the approximate cavity distribution using BP in combination with linear response and to assume higher order cumulants to be zero (although in principle one could use higher order linear response estimates for the higher order cumulants).

Here, instead, we propose to initialize the approximate cavity distributions by using standard BP on a "clamped" network. This means that for each cavity variable i, we fix some setting $x_{\partial i}$ of its Markov blanket, use BP to calculate the corresponding Bethe free energy $F_{Bethe}(x_{\partial i})$ for that particular setting, iterate over all possible settings, and finally calculate the approximate cavity distribution

$$Q_0^{\setminus i}(x_{\partial i}) \propto e^{-F_{Bethe}(x_{\partial i})}. \qquad (11)$$

In this way we capture all effective interactions, also higher order ones, in the initial cavity distributions.

One can think of many other ways to approximate the initial cavity distributions. The procedure described above is exponential in the size of the cavity. An alternative way of initiliazing the cavity distributions is to estimate the pair marginals $P^{\setminus i}(x_j, x_k)$ for each pair $(j,k) \in \partial i^2$. This can be done by clamping x_j to some value, using BP to approximate $P^{\setminus i}(x_k \mid x_j)$ and $F_{Bethe}(x_j)$. An approximation of $P^{\setminus i}(x_j, x_k)$ is then given by

$$q_0^{\setminus i}(x_j, x_k) := P^{\setminus i}(x_k \mid x_j) e^{-F_{Bethe}(x_j)}.$$

The approximate cavity distribution $Q_0^{\setminus i}$ is then simply the product of all approximated pair marginals:

$$Q^{\setminus i}(x_{\partial i}) \propto \prod_{\substack{\{j,k\} \\ j,k \in \partial i}} q_0^{\setminus i}(x_j, x_k) \qquad (12)$$

Algorithm 1 LCBP update algorithm (parallel updates)

```
$t \leftarrow 0$
repeat
    for all $i, j \in \mathcal{V}$ s.t. $i, j \in I$ for some $I \in \mathcal{F}$ do
        $Q_{t+1}^{\setminus j} \propto Q_t^{\setminus j} \frac{\sum_{x_{\partial i}} Q_t^{\setminus i} \Psi_i^{\setminus I}}{\sum_{x_{\Delta j \setminus i}} Q_t^{\setminus j} \Psi_j^{\setminus I}}$
    end for
    $t \leftarrow t + 1$
until convergence
```

This procedure is quadratic in cavity size. However, the update equations are still exponential in the cavity size.

As a side note, one can show that by simply taking completely factorized initial cavity distributions (i.e. $Q_0^{\setminus i}(x_{\partial i}) \propto \prod_{j \in \partial i} q_j^{\setminus i}(x_j)$ for arbitrary $q_j^{\setminus i}$), fixed points of BP are fixed points of Algorithm 1. Thus LCBP can indeed be regarded as a loop correction scheme for the Bethe approximation.

2.6 EXACTNESS IN CASE OF ONE LOOP

It was shown in [1] that the method proposed there is exact if the graphical model contains only one loop, possibly attached to treelike structures. Using a similar argument, we can show that a similar result holds for our alternative method. Suppose the graphical model contains exactly one loop. Consider first the case that i is part of the loop; removing i will break the loop and the remaining cavity graph will be singly connected, hence the cavity distribution calculated by BP will be exact. On the other hand, if i is not part of the loop, removing i will divide the network into several connected components, one for each neighbour of i; this implies that the cavity distribution calculated by BP contains no higher order interactions, i.e. $Q_0^{\setminus i}$ is exact modulo single node interactions. Hence, after running the LCBP update algorithm, all cavity distributions will be exact, which obviously implies that the final single node marginals will be exact.

3 EXPERIMENTS

We have performed numerical experiments to compare the quality of the results and the computation time of the following approximate inference methods:

BP Standard BP, using the recently proposed update scheme [12], which converges also for difficult problems without damping.

HAK-Δ A double-loop implementation [13] of CVM using the sets $\{\Delta i\}_{i \in \mathcal{V}}$ as outer clusters.[4]

HAK-4 A double-loop implementation of CVM using as outer clusters all factors together with all loops in the factor graph that consist of up to 4 different variables.

TreeEP TreeEP [8], without damping.

[4] We have used a double-loop implementation of CVM instead of GBP because the former is guaranteed to convergence to a local minimum of the Kikuchi free energy [13], whereas the latter often only would converge with strong damping, where the required damping constant is not known *a priori*.

LCBP-CUM The original cumulant-based loop correction scheme described in [1].

LCBP LCBP with cavities initialized as in (11).

LCBP-PAIR LCBP with cavities initialized as in (12).

To be able to assess the errors of the various approximate methods, we limited ourselves to problems for which exact inference (using a standard junction tree method) was still feasible.

For each approximate inference method, we have calculated the maximum error in the approximate single node marginals q_i as follows:

$$\max_{i \in \mathcal{V}} \max_{x_i \in \mathcal{X}_i} |q_i(x_i) - p_i(x_i)| \tag{13}$$

where $p_i(x_i) = P(x_i)$ is the exact marginal.[5]

The computation time was measured as CPU time in seconds on a 2.4 GHz AMD Opteron 64bits processor with 4 GB memory. The timings should be seen as indicative, as we have only optimized BP. The implementations of the other approximate inference can still be optimized for speed, which may alter the timings reported here by some constant depending on the method.[6]

We have studied three different model classes: (i) random graphs with fixed degree $d = 5$ and binary variables; (ii) periodic square grids with binary variables; (iii) the ALARM network.

3.1 RANDOM REGULAR GRAPHS WITH BINARY VARIABLES

We have compared various approximate inference methods on random graphs with fixed degree $|\partial i| = 5$ with ± 1-valued variables. Random graphs are special in the sense that the number of short loops is relatively small. As single node factors we took $\psi_i(x_i) = \exp(\theta_i x_i)$ for i.i.d. weights θ_i drawn from a $\mathcal{N}(0, \beta)$. For the pairwise factors we took $\psi_{ij}(x_i, x_j) = \exp(J_{ij} x_i x_j)$ for i.i.d. weights J_{ij}, also drawn from a $\mathcal{N}(0, \beta)$ distribution. The parameter β controls the strength of the interactions and the difficulty of the inference problem.

Figure 2 shows the results for $\beta = 1$. LCBP is the most accurate method. BP is the fastest method but is not very accurate. HAK-Δ performs remarkably bad, being the slowest and the least accurate method of all. This is remarkable, since one would expect that it should at least improve on BP since it uses larger clusters. It shows that although both LCBP and HAK-Δ use identical clusters, the nature of both approximations is very different. TreeEP is more accurate than BP but still very efficient in terms of computation time. LCBP almost always significantly improves upon BP, often more than one order of magnitude, except in cases where the BP error itself is of order 1. For $N = 70$, the treesize became so large that exact inference was infeasible. Although LCBP-CUM and LCBP-PAIR use rather similar initial approximate cavity distributions, their end results differ significantly. In this regime of strong interactions, LCBP-CUM is seen not to improve on BP.

For smaller β, i.e. weaker interactions, the results become more pronounced; in particular, the relative improvement of LCBP over TreeEP and BP increases for weaker interactions. For stronger interactions (higher β), performance of the various methods becomes more instance dependent, but on average the picture remains the same. The only exception is LCBP-CUM, which performs better for smaller β, comparable to LCBP-PAIR (not shown).

3.2 PERIODIC SQUARE GRIDS

The next class of models are periodic square grids (i.e. square grids on a torus) with binary variables. These models have many short loops, making them difficult problems for approximate inference. The special topology of these graphical models allows for a natural choice of the outer clusters for CVM, namely 2×2 plaquettes. Thus in addition to HAK-Δ (which in this case uses +-shaped clusters consisting of 5 variables), we compare with HAK-4, a double-loop implementation of CVM using the 2×2 plaquettes. We took the same kind of interactions as for the random graphs.

The results can be found in Figure 3. HAK-Δ was so slow that we did not consider it. As for random graphs, the fastest method is BP. As for the random graphs, TreeEP improves significantly on BP, using little computation time. Again LCBP uses more computation time but improves the accuracy even more. The HAK-4 method shows a surprising behaviour: for small grids, its accuracy is comparable to that of BP, whereas for larger grids, the accuracy quickly improves and it becomes the most accurate of all tested methods. Note that the treewidth quickly increases with N and for large grids computation time for exact inference is comparable to that of the slowest approximate inference methods.

[5] We have considered other error measures as well (average maximum single node error, maximum and average Kullback-Leibler divergence). We do not report these results here because of space constraints and because the choice of error measure does not affect our conclusions.

[6] We plan to release our C++ implementation of the various algorithms as free software.

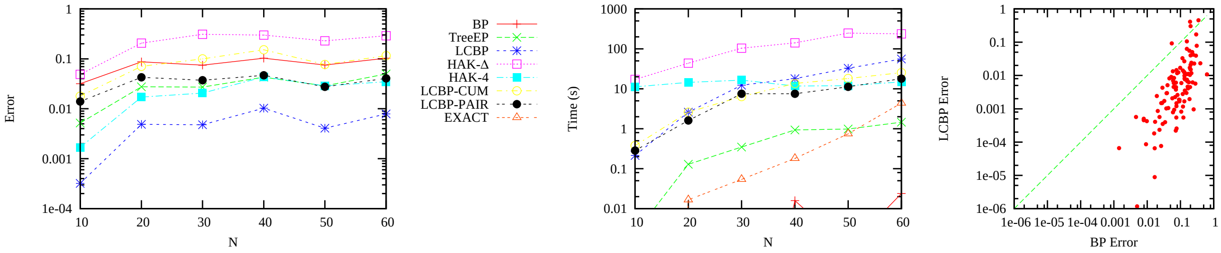


Figure 2: Errors and computation times for random graphs with degree 5 and interaction strength $\beta = 1$. Left: errors of single node marginals vs. graph size. Middle: computation time vs. graph size. Right: LCBP error vs. BP error. Each point in the left and middle plots is an average (in the log-domain) over 10 randomly generated instances.

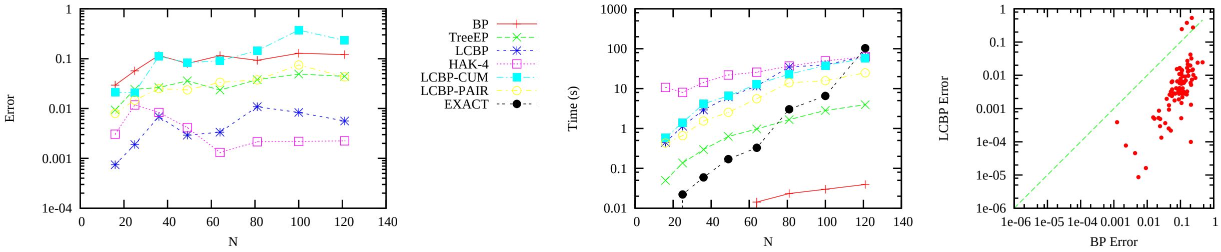


Figure 3: Periodic square grids, $\beta = 1$. Left: errors of single node marginals vs. graph size. Middle: computation time vs. graph size. Right: LCBP error vs. BP error. Each point in the left and middle plots is an average (in the log-domain) over 10 randomly generated instances.

3.3 ALARM NETWORK

The ALARM network is a well-known Bayesian network consisting of 37 variables and higher order factors. In addition to the usual approximate inference methods, we have compared with GBP, using maximal factors as outer clusters. The results are reported in Table 1.

The accuracy of GBP is almost identical to that of BP on this model. Again we see that improving the cluster size (HAK-Δ) does not improve the results, it even makes them worse. LCBP reduces the error by a factor of about 400. Unfortunately, computation time increases even more. An important direction for future research would be to extend and generalize the loop correction framework in order to find different tradeoffs between computation time and accuracy.

4 DISCUSSION AND CONCLUSIONS

We have proposed a method for improving BP by taking into account all the loops in the factor graph, which is a variation of the one proposed in [1]. We have shown that it can significantly outperform other approximate inference methods in terms of accuracy. On the downside, the computation time is rather high and application is limited to graphical models with small cavities. Further we have shown that simply increasing the cluster size in CVM (GBP) does not guarantee better results. In fact, often the results were even worse than for the simplest cluster choice (i.e. the outer clusters being the maximal factors, which coincides with BP in case of pairwise factors). Because LCBP and HAK-Δ use identical clusters, one might think naïvely that both approximation method will behave similarly; however, as we have shown, this is not the case, and the nature of both approximations appears to be completely different.

In almost all cases, the LCBP results had significantly smaller errors than the BP results, the only exceptions being the cases where the BP error was of order 1. In most cases, LCBP results were also significantly more accurate than TreeEP.

Only for grids we encountered an approximate inference method that appears to be structurally better than LCBP. Here, CVM with 2×2 plaquettes outperforms LCBP. A possible explanation may be that in this case the shortest and most important loops are included in an outer cluster each (although this does not explain why for small grids the error is comparable to that of BP, which assumes no loops).

For large graphs, exact inference can become intractable and in these cases, LCBP may be a viable alternative (provided the cavities are small) that in our experience often gives highly accurate results.

The most important difference between the method proposed here and the original one in [1] is that we assume that the cavity distributions contain no higher order *interactions* (i.e. interactions involving more than two cavity variables), whereas the original proposal is to assume that higher order *cumulants* vanish. Both approaches are identical to first order in the corrections $\phi_j^{\setminus i}(x_j)$. However, the cumulant-based formulation has several disadvantages. First, it is difficult to work with in practice, because it leads to rather complicated expressions. Further, it is not obvious how to generalize it beyond the binary, pairwise case, although this should be possible in principle. Finally, the approximation of vanishing higher order cumulants turns out to break down in the regime of strong interactions, whereas our interaction-based approximation still works in that regime.

Concluding, the LCBP method proposed in this work appears to be ideally suited to compute with high accuracy single node marginals for graphical models having small cavities, especially when the graph has "long" loops which cannot easily be taken into account exactly using CVM or other region-based methods. On these graphical models, the quality of the results turned out to be superior to the other approximate inference methods we compared with.

There still appears to be room for improvement of the LCBP method as formulated here. In particular, various alternatives to the LCBP update equations (line 4 of Algorithm 1) are possible and can give even better results. As an example, consider altered update equations in which the connecting factor ψ_I is *not* divided out (equivalent to demanding consistency of single node marginals for the original, unmodified, probability distribution (1)). This does not significantly alter the results for weak, pairwise factors, but appears to be more robust if the factors are stronger. On the other hand, in the presence of factors involving more than two variables, this alternative approach leads to significantly worse results. This observation suggests the possible existence of update equations in the same spirit as line 4 in Algorithm 1, but which give better results in general. We intend to address this issue in more detail in a forthcoming publication.

Table 1: Results for the ALARM network. Apart from the error measure (13) ("Max MAD"), we also report the average maximum absolute deviation of the single node marginals ("Avg MAD").

Method	Time (s)	Max MAD	Avg MAD
BP	0.01	0.203	0.00814
GBP	0.18	0.203	0.00759
TreeEP	0.04	0.039	0.011
HAK-Δ	293.14	0.223	0.0738
LCBP	14.62	0.000544	0.0000151

Acknowledgements

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024.

References

[1] Andrea Montanari and Tommaso Rizzo. How to compute loop corrections to the Bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10011, 2005.

[2] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized Belief Propagation. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 13 (NIPS*2000)*, pages 689–695, Cambridge, MA, 2001. MIT Press.

[3] H. Bethe. Statistical theory of superlattices. *Proc. R. Soc. A*, 150:552–575, 1935.

[4] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and Generalized Belief Propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, July 2005.

[5] A. Pelizzola. Cluster variation method in statistical physics and probabilistic graphical models. *J. Phys. A: Math. Gen.*, 38:R309–R339, August 2005.

[6] R. Kikuchi. A theory of cooperative phenomena. *Phys. Rev.*, 81:988–1003, 1951.

[7] Thomas Minka. Expectation Propagation for approximate Bayesian inference. In *Proc. of the 17th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-01)*, pages 362–369, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[8] Thomas Minka and Yuan Qi. Tree-structured approximations by Expectation Propagation. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[9] Giorgio Parisi and Frantisek Slanina. Loop expansion around the Bethe-Peierls approximation for lattice models. *arXiv.org preprint*, cond-mat/0512529, 2005.

[10] Michael Chertkov and Vladimir Y. Chernyak. Loop series for discrete statistical models on graphs. *arXiv.org preprint*, cond-mat/0603189, 2006.

[11] H.-O. Georgii. *Gibbs Measures and Phase Transitions*. Walter de Gruyter, Berlin, 1988.

[12] G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, Boston, Massachussetts, July 2006.

[13] Tom Heskes, C.A. Albers, and Hilbert J. Kappen. Approximate inference and constrained optimization. In *Proc. of the 19th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-03)*, pages 313–320, San Francisco, CA, 2003. Morgan Kaufmann Publishers.