Loop corrections for approximate inference

Joris Mooij j.mooij@science.ru.nl
Department of Biophysics
Radboud University Nijmegen
6525 EZ Nijmegen, The Netherlands

Bert Kappen b.kappen@science.ru.nl
Department of Biophysics
Radboud University Nijmegen
6525 EZ Nijmegen, The Netherlands

Editor:

Abstract

We propose a method for improving approximate inference methods that corrects for the influence of loops in the graphical model. The method is applicable to arbitrary factor graphs, provided that the size of the Markov blankets is not too large. It is an alternative implementation of an idea introduced recently by Montanari and Rizzo (2005). In its simplest form, which amounts to the assumption that no loops are present, the method reduces to the minimal Cluster Variation Method approximation (which uses maximal factors as outer clusters). On the other hand, using estimates of the effect of loops (obtained by some approximate inference algorithm) and applying the Loop Correcting (LC) method usually gives significantly better results than applying the approximate inference algorithm directly without loop corrections. Indeed, we often observe that the loop corrected error is approximately the square of the error of the approximate inference method used to estimate the effect of loops. We compare different variants of the Loop Correcting method with other approximate inference methods on a variety of graphical models, including "real world" networks, and conclude that the LC approach generally obtains the most accurate results.

Keywords: Loop Corrections, Approximate Inference, Graphical Models, Factor Graphs

1. Introduction

In recent years, much research has been done in the field of approximate inference on graphical models. One of the goals is to obtain accurate approximations of marginal probabilities of complex probability distributions defined over many variables, using limited computation time and memory. This research has led to a large number of approximate inference methods. Apart from sampling ("Monte Carlo") methods, the most well-known methods and algorithms are variational approximations such as Mean Field (MF), which originates in statistical physics (Parisi, 1988); Belief Propagation (BP), also known as the Sum-Product Algorithm and as Loopy Belief Propagation (Pearl, 1988; Kschischang et al., 2001), which is directly related to the Bethe approximation used in statistical physics (Bethe, 1935; Yedidia et al., 2005); the Cluster Variation Method (CVM) (Pelizzola, 2005) and other region-based

approximation methods (Yedidia et al., 2005), which are related to the Kikuchi approximation (Kikuchi, 1951), a generalization of the Bethe approximation using larger clusters; Expectation Propagation (EP) (Minka, 2001), which includes TreeEP (Minka and Qi, 2004) as a special case. To calculate the results of CVM and other region based approximation methods, one can use the Generalized Belief Propagation (GBP) algorithm (Yedidia et al., 2005) or double-loop algorithms that have guaranteed convergence (Yuille, 2002; Heskes et al., 2003).

It is well-known that Belief Propagation yields exact results if the graphical model is a tree, or, more generally, if each connected component is a tree. If the graphical model does contain loops, BP can still yield surprisingly accurate results using little computation time. However, if the influence of loops is large, the approximate marginals calculated by BP can have large errors and the quality of the BP results may not be satisfactory. One way to correct for the influence of short loops is to increase the cluster size of the approximation, using CVM (GBP) with clusters that subsume as many loops as possible. However, choosing a good set of clusters is highly nontrivial (Welling et al., 2005), and in general this method will only work if the clusters do not have many intersections, or in other words, if the loops do not have many intersections. Another method that corrects for loops to a certain extent is TreeEP, which does exact inference on the base tree, a subgraph of the graphical model which has no loops, and approximates the other interactions. This corrects for the loops that consist of part of the base tree and exactly one additional factor and yields good results if the graphical model is dominated by the base tree, which is the case in very sparse models. However, loops that consist of two or more interactions that are not part of the base tree are approximated in a similar way as in BP. Hence, for denser models, the improvement of TreeEP over BP usually diminishes.

In this article we propose a method that takes into account *all* the loops in the graphical model in an approximate way and therefore obtains more accurate results in many cases. Our method is a variation on the theme introduced by Montanari and Rizzo (2005). The basic idea is to first estimate the cavity distributions of all variables and subsequently improve these estimates by cancelling out errors using certain consistency constraints. A cavity distribution of some variable is the probability distribution on its Markov blanket (all its neighbouring variables) of a modified graphical model, in which all factors involving that variable have been removed. The removal of the factors breaks all the loops in which that variable takes part. This allows an approximate inference algorithm to estimate the strength of these loops in terms of effective interactions or correlations between the variables of the Markov blanket. Then, the influence of the removed factors is taken into account, which yields accurate approximations to the probability distributions of the original graphical model. Even more accuracy is obtained by imposing certain consistency relations between the cavity distributions, which results in a cancellation of errors to some extent. This error cancellation is done by a message passing algorithm which can be interpreted as a generalization of BP in the pairwise case and of the minimal CVM approximation in general. Indeed, the assumption that no loops are present, or equivalently, that the cavity distributions factorize, yields the BP / minimal CVM results. On the other hand, using better estimates of the effective interactions in the cavity distributions yields accurate loop corrected results.

Although the basic idea underlying our method is very similar to that described in (Montanari and Rizzo, 2005), the alternative implementation that we propose here offers two advantages. Most importantly, it is directly applicable to arbitrary factor graphs, whereas the original method has only been formulated for the rather special case of graphical models with binary variables and pairwise factors, which excludes e.g. many interesting Bayesian networks. Furthermore, our implementation appears to be more robust and also gives improved results for relatively strong interactions, as will be shown numerically.

This article is organised as follows. First we explain the theory behind our proposed method and discuss the differences with the original method by Montanari and Rizzo (2005). Then we report extensive numerical experiments regarding the quality of the approximation and the computation time, where we compare with other approximate inference methods. Finally, we discuss the results and state conclusions.

2. Theory

In this work, we consider graphical models such as Markov random fields and Bayesian networks. We use the general factor graph representation since it allows for formulating approximate inference algorithms in a unified way (Kschischang et al., 2001). In the next subsection, we introduce our notation and basic definitions.

2.1 Graphical models and factor graphs

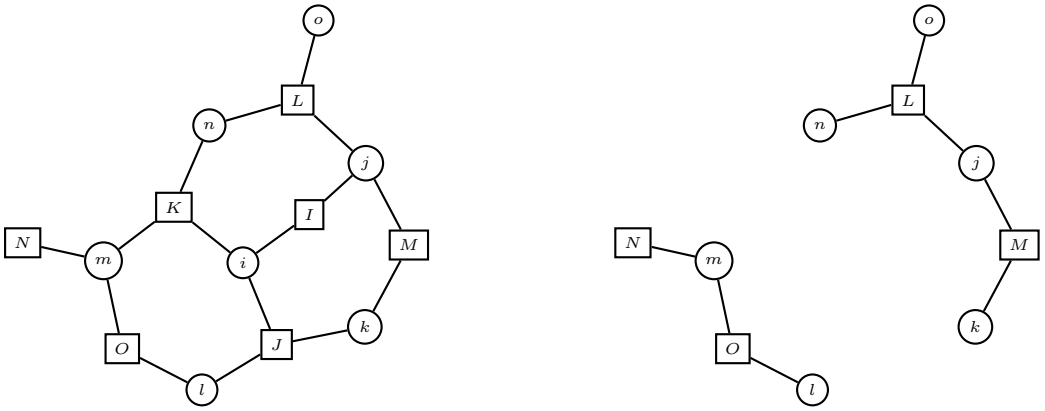
Consider N discrete random variables $\{x_i\}_{i\in\mathcal{V}}$ with $\mathcal{V} := \{1,\dots,N\}$. Each variable x_i takes values in a discrete domain $\mathcal{X}_i$. We will use the following multi-index notation: for any subset $I \subseteq \mathcal{V}$, we write $x_I := (x_{i_1}, x_{i_2}, \dots,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

$$P(x) = \frac{1}{Z}\prod_{I\in\mathcal{F}} \psi_I(x_I), \qquad Z = \sum_x \prod_{I\in\mathcal{F}} \psi_I(x_I). \tag{1}$$

The factors (which we will also call "interactions") are indexed by (small) subsets of $\mathcal{V}$, i.e. $\mathcal{F} \subseteq \mathcal{P}(\mathcal{V}) := \{I : I \subseteq \mathcal{V}\}$. Each factor is a non-negative function $\psi_I : \prod_{i\in I} \mathcal{X}_i \to [0,\infty)$. For a Bayesian network, the factors are conditional probability tables. In case of Markov random fields, the factors are often called potentials (not to be confused with statistical physics terminology, where "potential" refers to minus the logarithm of the factor instead). Henceforth, we will refer to a triple $(\mathcal{V}, \mathcal{F}, \{\psi_I\}_{I\in\mathcal{F}})$ that satisfies the description above as a discrete *graphical model* (or *network*).

In general, the normalizing constant Z is not known and exact computation of Z is infeasible, due to the fact that the number of terms to be summed is exponential in N. Similarly, computing marginal distributions $P(x_J)$ of P for subsets of variables $J \subseteq \mathcal{V}$ is intractable in general. In this article, we focus on the task of accurately approximating single node marginals $P(x_i) = \sum_{x_{\mathcal{V}\setminus i}} P(x)$.

We can represent the structure of the probability distribution (1) using a *factor graph*. This is a bipartite graph, consisting of *variable nodes* $i \in \mathcal{V}$ and *factor nodes* $I \in \mathcal{F}$, with an edge between i and I if and only if $i \in I$, i.e. if x_i participates in the factor ψ_I. We



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

Figure 1: (a) Original factor graph, corresponding to the probability distribution $P(x) = \frac{1}{Z}\psi_L(x_j,x_n,x_o)\psi_I(x_i,x_j)\psi_M(x_j,x_k)\psi_N(x_m)\psi_K(x_i,x_m,x_n)\psi_J(x_i,x_k,x_l)\psi_O(x_l,x_m)$; (b) Factor graph corresponding to the cavity network of variable i, obtained by removing variable i and the factor nodes that contain i (i.e. I,J and K). The Markov blanket of i is $\partial i = \{j,k,l,m,n\}$. The cavity distribution $Z^{\backslash i}(x_{\partial i})$ is the (unnormalized) marginal on $x_{\partial i}$ of the probability distribution corresponding to the cavity graph (b).

will represent factor nodes visually as rectangles and variable nodes as circles. See Figure 1(a) for an example of a factor graph. We denote the neighbouring nodes of a variable node i by $N_i := \{I \in \mathcal{F} : i \in I\}$ and the neighbouring nodes of a factor node I simply by $I = \{i \in \mathcal{V} : i \in I\}$. Further, we define for each variable $i \in \mathcal{V}$ the set $\Delta i := \bigcup N_i$ consisting of all variables that appear in some factor in which variable i participates, and the set $\partial i := \Delta i \setminus \{i\}$, the *Markov blanket* of i.

In the following, we will often abbreviate the set theoretical notation $X \setminus Y$ (i.e. all elements in X that are not in Y) by $\setminus Y$ if it is obvious from the context what the set X is. Further, we will use lowercase for variable indices and uppercase for factor indices. For convenience, we will define for any subset $\mathcal{I} \subset \mathcal{F}$ the product of the corresponding factors:

$$\Psi_{\mathcal{I}}(x_{\bigcup \mathcal{I}}) := \prod_{I \in \mathcal{I}} \psi_I(x_I).$$

2.2 Cavity networks and loop corrections

The notion of a *cavity* stems from statistical physics, where it was used originally to calculate properties of random ensembles of certain graphical models (Mézard et al., 1987). A cavity is obtained by removing one variable from the graphical model, together with all the factors in which that variable participates.

In our context, we define cavity networks as follows (see also Figure 1):

Definition 2.1 *Given a graphical model $(\mathcal{V},\mathcal{F},\{\psi_I\}_{I\in\mathcal{F}})$ and a variable $i \in \mathcal{V}$, the* cavity network *of variable i is the graphical model $(\mathcal{V}\setminus i, \mathcal{F}\setminus N_i, \{\psi_I\}_{I\in\mathcal{F}\setminus N_i})$.*

The probability distribution corresponding to the cavity network of variable i is thus proportional to:

$$\Psi_{\backslash N_i}(x_{\backslash i}) = \prod_{\substack{I \in \mathcal{F} \\ i \notin I}} \psi_I(x_I).$$

Summing out all the variables, except for the neighbours ∂i of i, gives what we will call the *cavity distribution*:

Definition 2.2 *Given a graphical model $(\mathcal{V}, \mathcal{F}, \{\psi_I\}_{I \in \mathcal{F}})$ and a variable $i \in \mathcal{V}$, the* cavity distribution *of i is*

$$Z^{\backslash i}(x_{\partial i}) := \sum_{x_{\backslash \Delta i}} \Psi_{\backslash N_i}(x_{\backslash i}). \tag{2}$$

Thus the cavity distribution of i is proportional to the marginal of the cavity network of i on the Markov blanket ∂i. The cavity distribution describes the *effective* interactions (or correlations) induced by the cavity network on the neighbours ∂i of variable i. Indeed, from equations (1) and (2) and the trivial observation that $\Psi_{\mathcal{F}} = \Psi_{N_i} \Psi_{\backslash N_i}$ we conclude:

$$P(x_{\Delta i}) \propto Z^{\backslash i}(x_{\partial i}) \Psi_{N_i}(x_{\Delta i}). \tag{3}$$

Thus given the cavity distribution $Z^{\backslash i}(x_{\partial i})$, one can calculate the marginal distribution of the original graphical model P on $x_{\Delta i}$, provided that the cardinality of $\mathcal{X}_{\Delta i}$ is not too large.

In practice, exact cavity distributions are not known, and the only way to proceed is to use approximate cavity distributions. Given some approximate inference method (e.g. BP), there are two ways to calculate $P(x_{\Delta i})$: either use the method to approximate $P(x_{\Delta i})$ directly, or use the method to approximate $Z^{\backslash i}(x_{\partial i})$ and use relation (3) to obtain an approximation to $P(x_{\Delta i})$. The latter method generally gives more accurate results, since the complexity of the cavity network is less than that of the original network. In particular, the cavity network of variable i contains no loops involving that variable, since all factors in which i participates have been removed (e.g. the loop $i - J - l - O - m - K - i$ in the original network, Figure 1(a), is not present in the cavity network, Figure 1(b)). Thus the latter method of calculating $P(x_{\Delta i})$ takes into account loops involving variable i, although in an approximate way. It does not, however, take into account the other loops in the original graphical model. The basic idea of the loop correction approach of Montanari and Rizzo (2005) is to use the latter method for all variables in the network, but to adjust the approximate cavity distributions in order to cancel out approximation errors before (3) is used to obtain the final approximate marginals. This approach takes into account *all* the loops in the original network, in an approximate way.

This basic idea can be implemented in several ways. Here we propose an implementation which we will show to have certain advantages over the original implementation proposed in (Montanari and Rizzo, 2005). In particular, it is directly applicable to arbitrary factor graphs with variables taking an arbitrary (discrete) number of values and factors that may contain zeroes and consist of an arbitrary number of variables. In the remaining subsections, we will first discuss our proposed implementation in detail. In section 2.6 we will discuss differences with the original approach.

2.3 Combining approximate cavity distributions to cancel out errors

Suppose that we have obtained an initial approximation $\zeta_0^{\backslash i}(x_{\partial i})$ of the (exact) cavity distribution $Z^{\backslash i}(x_{\partial i})$, for each $i \in \mathcal{V}$. Let $i \in \mathcal{V}$ and consider the approximation error of the cavity distribution of i, i.e. the exact cavity distribution of i divided by its approximation:

$$\frac{Z^{\backslash i}(x_{\partial i})}{\zeta_0^{\backslash i}(x_{\partial i})}.$$

In general, this is an arbitrary function of the variables $x_{\partial i}$. However, for our purposes, we can *approximate* the error as a product of factors defined on small subsets of ∂i in the following way:

$$\frac{Z^{\backslash i}(x_{\partial i})}{\zeta_0^{\backslash i}(x_{\partial i})} \approx \prod_{I \in N_i} \phi_I^{\backslash i}(x_{I \backslash i}).$$

Thus we assume that the approximation error lies near a submanifold parameterized by the error factors $\{\phi_I^{\backslash i}(x_{I\backslash i})\}_{I \in N_i}$. If we were able to calculate these error factors, we could improve our initial approximation $\zeta_0^{\backslash i}(x_{\partial i})$ by replacing it with the product

$$\zeta^{\backslash i}(x_{\partial i}) := \zeta_0^{\backslash i}(x_{\partial i}) \prod_{I \in N_i} \phi_I^{\backslash i}(x_{I \backslash i}) \approx Z^{\backslash i}(x_{\partial i}). \tag{4}$$

Using (3), this would then yield an improved approximation of $P(x_{\Delta i})$.

It turns out that the error factors can indeed be calculated by exploiting the redundancy of the information in the initial cavity approximations $\{\zeta_0^{\backslash i}\}_{i \in \mathcal{V}}$. The fact that all $\zeta^{\backslash i}$ provide approximations to marginals of the *same* probability distribution $P(x)$ via (3) can be used to obtain consistency constraints. The number of constraints obtained in this way is enough to solve for the unknown error factors $\{\phi_I^{\backslash i}(x_{I\backslash i})\}_{i \in \mathcal{V}, I \in N_i}$.

Here we propose the following consistency constraints. Let $Y \in \mathcal{F}$, $i \in Y$ and $j \in Y$ with $i \neq j$ (see also Figure 2). Consider the graphical model $(\mathcal{V}, \mathcal{F} \setminus Y, \{\psi_I\}_{I \in \mathcal{F} \setminus Y})$ that is obtained from the original graphical model by removing factor ψ_Y. The product of all factors (except ψ_Y) obviously satisfies:

$$\Psi_{\backslash Y} = \Psi_{N_i \backslash Y} \Psi_{\backslash N_i} = \Psi_{N_j \backslash Y} \Psi_{\backslash N_j}.$$

Using (2) and summing over all x_k for $k \notin Y \setminus i$, we obtain the following equation, which holds for the exact cavity distributions $Z^{\backslash i}$ and $Z^{\backslash j}$:

$$\sum_{x_i} \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i \backslash Y} Z^{\backslash i} = \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} Z^{\backslash j}.$$

Substituting our basic assumption (4) on both sides and pulling the factor $\phi_Y^{\backslash i}(x_{Y \backslash i})$ in the l.h.s. through the summation, we obtain:

$$\phi_Y^{\backslash i} \sum_{x_i} \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i \backslash Y} \zeta_0^{\backslash i} \prod_{I \in N_i \backslash Y} \phi_I^{\backslash i} = \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} \zeta_0^{\backslash j} \prod_{J \in N_j} \phi_J^{\backslash j} \tag{5}$$

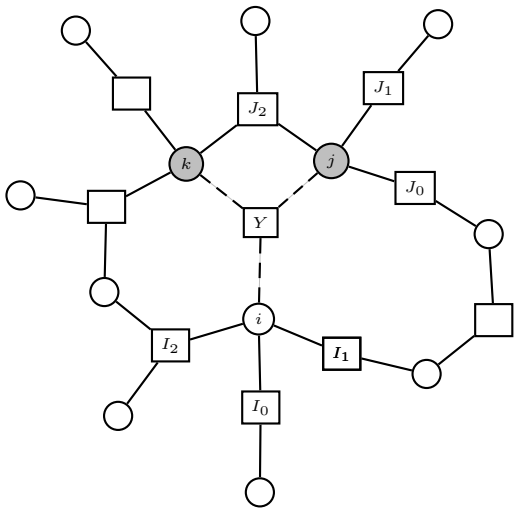


Figure 2: Part of the factor graph, illustrating the derivation of (6). The two grey variable nodes correspond to $Y \setminus i = \{j, k\}$.

This should hold for each $j \in Y \setminus i$. We can thus take the geometrical average of the r.h.s. over all $j \in Y \setminus i$. After rearranging, this yields:

$$\phi_Y^{\setminus i} = \frac{\left(\prod_{j \in Y \setminus i} \sum_{x_i} \sum_{x_{\Delta j \setminus Y}} \Psi_{N_j \setminus Y} \zeta_0^{\setminus j} \prod_{J \in N_j} \phi_J^{\setminus j} \right)^{1/|Y \setminus i|}}{\sum_{x_i} \sum_{x_{\Delta i \setminus Y}} \Psi_{N_i \setminus Y} \zeta_0^{\setminus i} \prod_{I \in N_i \setminus Y} \phi_I^{\setminus i}} \qquad \text{for all } i \in \mathcal{V},\, Y \in N_i. \tag{6}$$

Note that the numerator is an approximation of the joint marginal $P^{\setminus Y}(x_{Y \setminus i})$ of the modified graphical model $(\mathcal{V}, \mathcal{F} \setminus Y, \{\psi_I\}_{I \in \mathcal{F} \setminus Y})$ on the variables $Y \setminus i$.

Solving the consistency equations (6) simultaneously for the error factors $\{\phi_I^{\setminus i}\}_{i \in \mathcal{V}, I \in N_i}$ can be done using a simple fixed point iteration algorithm, e.g. Algorithm 1. The input consists of the initial approximations $\{\zeta_0^{\setminus i}\}_{i \in \mathcal{V}}$ to the cavity distributions. It calculates the error factors that satisfy (6) by fixed point iteration and from the fixed point, it calculates improved approximations of the cavity distributions $\{\zeta^{\setminus i}\}_{i \in \mathcal{V}}$ using relation (4). [1] From the improved cavity distributions, we can calculate the loop corrected approximations to the single variable marginals of the original probability distribution (1) as follows:

$$b_i(x_i) \propto \sum_{x_{\partial i}} \Psi_{N_i}(x_{\Delta i}) \zeta^{\setminus i}(x_{\partial i}) \tag{7}$$

where the factor ψ_Y is now included. Algorithm 1 uses a sequential update scheme, but other update schemes are possible (e.g. random sequential or parallel). In practice, the fixed sequential update scheme often converges without the need for damping.

1. Alternatively, one could formulate the updates directly in terms of the cavity distributions $\{\zeta^{\setminus i}\}$.

Algorithm 1 Loop Correcting algorithm

Input: initial approximate cavity distributions $\{\zeta_0^{\setminus i}\}_{i\in\mathcal{V}}$
Output: improved approximate cavity distributions $\{\zeta^{\setminus i}\}_{i\in\mathcal{V}}$

1: **repeat**
2: **for all** $i \in \mathcal{V}$ **do**
3: **for all** $Y \in N_i$ **do**
4: $$\phi_Y^{\setminus i}(x_{Y\setminus i}) \leftarrow \frac{\left(\prod_{j\in Y\setminus i}\sum_{x_i}\sum_{x_{\Delta j\setminus Y}}\Psi_{N_j\setminus Y}\zeta_0^{\setminus j}\prod_{J\in N_j}\phi_J^{\setminus j}\right)^{1/|Y\setminus i|}}{\sum_{x_i}\sum_{x_{\Delta i\setminus Y}}\Psi_{N_i\setminus Y}\zeta_0^{\setminus i}\prod_{I\in N_i\setminus Y}\phi_I^{\setminus i}}$$
5: **end for**
6: **end for**
7: **until** convergence
8: **for all** $i \in \mathcal{V}$ **do**
9: $\zeta^{\setminus i}(x_{\partial i}) \leftarrow \zeta_0^{\setminus i}(x_{\partial i})\prod_{I\in N_i}\phi_I^{\setminus i}(x_{I\setminus i})$
10: **end for**

Alternatively, one can formulate Algorithm 1 in terms of the "beliefs"

$$Q_i(x_{\Delta i}) \propto \Psi_{N_i}(x_{\Delta i})\zeta_0^{\setminus i}(x_{\partial i})\prod_{I\in N_i}\phi_I^{\setminus i}(x_{I\setminus i}) = \Psi_{N_i}(x_{\Delta i})\zeta^{\setminus i}(x_{\partial i}). \tag{8}$$

As one easily verifies, the following update equation

$$Q_i \leftarrow Q_i \frac{\prod_{j\in Y\setminus i}\left(\sum_{x_{\Delta j\setminus(Y\setminus i)}} Q_j\,\psi_Y^{-1}\right)^{1/|Y\setminus i|}}{\sum_{x_{\Delta i\setminus(Y\setminus i)}} Q_i\,\psi_Y^{-1}}$$

is equivalent to line 1 of Algorithm 1. Intuitively, the update improves the approximate distribution Q_i on Δi by replacing its marginal on $Y \setminus i$ (in the absence of Y) by a more accurate approximation of this marginal, namely the numerator. Written in this form, the algorithm is reminiscent of Iterative Proportional Fitting (IPF). However, contrary to IPF, the desired marginals are also updated each iteration. Note that after convergence, the large beliefs $Q_i(x_{\Delta i})$ need not be consistent, i.e. in general $\sum_{x_{\Delta i\setminus J}} Q_i \neq \sum_{x_{\Delta j\setminus J}} Q_j$ for $i, j \in \mathcal{V}$, $J \subseteq \Delta i \cap \Delta j$.

2.4 A special case: factorized cavity distributions

In the previous subsection we have discussed how to improve approximations of cavity distributions. We now discuss what happens when we use the simplest possible initial approximations $\{\zeta_0^{\setminus i}\}_{i\in\mathcal{V}}$, namely constant functions, in Algorithm 1. This amounts to the assumption that no loops are present. We will show that if the factor graph does not contain

short loops consisting of four nodes, fixed points of the standard BP algorithm are also fixed points of Algorithm 1. In this sense, Algorithm 1 can be considered to be a generalization of the BP algorithm. In fact, this holds even if the initial approximations factorize in a certain way, as we will show below.

If all factors involve at most two variables, one can easily arrange for the factor graph to have no loops of four nodes. See figure 1(a) for an example of a factor graph which has no loops of four nodes. The factor graph depicted in Figure 2 does have a loop of four nodes: $k - Y - j - J_2 - k$.

Theorem 2.1 *If the factor graph corresponding to (1) has no loops of exactly four nodes, and all initial approximate cavity distributions factorize in the following way:*

$$\zeta_0^{\backslash i}(x_{\partial i}) = \prod_{I \in N_i} \xi_I^{\backslash i}(x_{I \backslash i}) \qquad \forall i \in \mathcal{V}, \tag{9}$$

then fixed points of the BP algorithm can be mapped to fixed points of Algorithm 1. Furthermore, the corresponding variable and factor marginals obtained from (8) are identical to the BP beliefs.

Proof Note that replacing the initial cavity approximations by

$$\zeta_0^{\backslash i}(x_{\partial i}) \mapsto \zeta_0^{\backslash i}(x_{\partial i}) \prod_{I \in N_i} \epsilon_I^{\backslash i}(x_{I \backslash i}) \tag{10}$$

for arbitrary positive functions $\epsilon_I^{\backslash i}(x_{I \backslash i})$ does not change the beliefs (8) corresponding to the fixed points of (6). Thus, without loss of generality, we can assume $\zeta_0^{\backslash i}(x_{\partial i}) = 1$ for all $i \in \mathcal{V}$. The BP update equations are:

$$\begin{aligned} \mu_{j \to I}(x_j) &\propto \prod_{J \in N_j \backslash I} \mu_{J \to j}(x_j) && j \in \mathcal{V}, I \in N_j \\ \mu_{I \to i}(x_i) &\propto \sum_{x_{I \backslash i}} \psi_I(x_I) \prod_{j \in I \backslash i} \mu_{j \to I}(x_j) && I \in \mathcal{F}, i \in I \end{aligned} \tag{11}$$

in terms of messages $\{\mu_{J \to j}(x_j)\}_{j \in \mathcal{V}, J \in N_j}$ and $\{\mu_{j \to J}(x_j)\}_{j \in \mathcal{V}, J \in N_j}$. Assume that the messages μ are a fixed point of (11) and take the *Ansatz*

$$\phi_I^{\backslash i}(x_{I \backslash i}) = \prod_{k \in I \backslash i} \mu_{k \to I}(x_k) \qquad \text{for } i \in \mathcal{V}, I \in N_i.$$

Then, for $i \in \mathcal{V}$, $Y \in N_i$, $j \in Y \setminus i$, we can write out part of the numerator of (6) as follows:

$$\begin{aligned} \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} \zeta_0^{\backslash j} \prod_{J \in N_j} \phi_J^{\backslash j} &= \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \phi_Y^{\backslash j} \prod_{J \in N_j \backslash Y} \psi_J \phi_J^{\backslash j} \\ &= \sum_{x_i} \left(\prod_{k \in Y \backslash j} \mu_{k \to Y} \right) \prod_{J \in N_j \backslash Y} \sum_{x_{J \backslash j}} \psi_J \prod_{k \in J \backslash j} \mu_{k \to J} \\ &= \sum_{x_i} \left(\prod_{k \in Y \backslash j} \mu_{k \to Y} \right) \mu_{j \to Y} = \sum_{x_i} \prod_{k \in Y} \mu_{k \to Y} \propto \prod_{k \in Y \backslash i} \mu_{k \to Y} \\ &= \phi_Y^{\backslash i}, \end{aligned}$$

where we used the BP update equations (11) and rearranged the summations and products using the assumption that the factor graph has no loops of four nodes. Thus, the numerator of the r.h.s. of (6) is simply $\phi_Y^{\backslash i}$. Using a similar calculation, one can derive that the denominator of the r.h.s. of (6) is constant, and hence equation (6) is valid (up to an irrelevant constant).

For $Y \in \mathcal{F}$, $i \in Y$, the marginal on x_Y of the belief (8) can be written in a similar way:

$$\begin{aligned}
\sum_{x_{\Delta i \backslash Y}} Q_i &\propto \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i} \prod_{I \in N_i} \phi_I^{\backslash i} = \sum_{x_{\Delta i \backslash Y}} \prod_{I \in N_i} \psi_I \prod_{k \in I \backslash i} \mu_{k \to I} \\
&= \psi_Y \left(\prod_{k \in Y \backslash i} \mu_{k \to Y} \right) \prod_{I \in N_i \backslash Y} \sum_{x_{I \backslash i}} \psi_I \prod_{k \in I \backslash i} \mu_{k \to I} \\
&= \psi_Y \left(\prod_{k \in Y \backslash i} \mu_{k \to Y} \right) \prod_{I \in N_i \backslash Y} \mu_{I \to i} = \psi_Y \left(\prod_{k \in Y \backslash i} \mu_{k \to Y} \right) \mu_{i \to Y} \\
&= \psi_Y \prod_{k \in Y} \mu_{k \to Y}.
\end{aligned}$$

which is proportional to the BP belief $b_Y(x_Y)$ on x_Y. Hence, also the single variable marginal b_i defined in (7) corresponds to the BP single variable belief, since both are marginalizations of b_Y for $Y \in N_i$. ■

If the factor graph does contain loops of four nodes, we find empirically that the fixed point of Algorithm 1, when using factorized initial cavity approximations as in (9), corresponds to the "minimal" CVM approximation, i.e. the CVM approximation that uses all (maximal) factors as outer clusters (Kikuchi, 1951; Pelizzola, 2005).[2] In that case, the factor beliefs found by Algorithm 1 are consistent, i.e. $\sum_{x_{\Delta i \backslash Y}} Q_i = \sum_{x_{\Delta j \backslash Y}} Q_j$ for $i, j \in Y$, and are identical to the minimal CVM factor beliefs. Thus it appears that Algorithm 1 can be considered as a generalization of the minimal CVM approximation (which can e.g. be calculated using the GBP algorithm (Yedidia et al., 2005) or a double-loop implementation (Heskes et al., 2003)).

We have not yet been able to prove this, so currently this claim stands as a conjecture, which we have empirically verified to be true for all the graphical models used for numerical experiments in section 3. Note that in case the factor graph has no loops of length four, the minimal CVM approximation reduces to the Bethe approximation, which yields a proof for this case. The proof in the general case is expected to be more involved, since one needs to keep track of various overlapping sets and it requires a translation of the structure of (6) (where the basic sets of variables involved are of three types, namely i, $Y \setminus i$, and Δi) and the GBP equations or Lagrange multiplier equations corresponding to the minimal CVM approximation (where the basic variable sets are those that can be written as an intersection of a finite number of factors).

2. Provided that the factor graph is connected.

2.5 Obtaining initial approximate cavity distributions

There is no principled way to obtain the initial cavity approximations $\zeta_0^{\setminus i}(x_{\partial i})$. In the previous subsection, we saw that factorized cavity approximations result in the minimal CVM approximation, which does not yet take into account the effect of loops in the cavity network. More sophisticated approximations that do take into account the effect of loops can significantly enhance the accuracy of the final result. In principle, there are many possibilities to obtain the initial cavity approximations. Here, we will describe one method, which uses BP on clamped cavity networks. This method captures all interactions in the cavity distribution of i in an approximate way and can lead to very accurate results. Instead of BP, any other approximate inference method that gives an approximation of the normalizing constant Z in (1) can be used, such as Mean Field, TreeEP (Minka and Qi, 2004), a double-loop version of BP (Heskes et al., 2003) which has guaranteed convergence towards a minimum of the Bethe free energy, or some variant of GBP (Yedidia et al., 2005). One could also choose the method for each cavity separately, trading accuracy versus computation time. We focus here on BP because it is a very fast and often relatively accurate algorithm.

Let $i \in \mathcal{V}$ and consider the cavity network of i. For each possible state of $x_{\partial i}$, run BP on the cavity network clamped to that state $x_{\partial i}$ and calculate the corresponding Bethe free energy $F_{Bethe}^{\setminus i}(x_{\partial i})$ (Yedidia et al., 2005). Take as initial approximate cavity distribution:

$$\zeta_0^{\setminus i}(x_{\partial i}) \propto e^{-F_{Bethe}^{\setminus i}(x_{\partial i})}.$$

This procedure is exponential in the size of ∂i: it uses $\prod_{j \in \partial i} |\mathcal{X}_j|$ BP runs. However, many networks encountered in applications are relatively sparse and have limited cavity size and the computational cost may be acceptable.

This particular way of obtaining initial cavity distributions has the following interesting property: in case the factor graph contains only a single loop, the final beliefs (8) resulting from Algorithm 1 are exact. This can be shown using an argument similar to that given in (Montanari and Rizzo, 2005). Suppose that the graphical model contains exactly one loop and let $i \in \mathcal{V}$. Consider first the case that i is part of the loop; removing i will break the loop and the remaining cavity network will be singly connected. The cavity distribution approximated by BP will thus be exact. Now 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. $\zeta_0^{\setminus i}$ is exact modulo single variable interactions. Because the final beliefs (8) are invariant under perturbation of the $\zeta_0^{\setminus i}$ by single variable interactions, the final beliefs calculated by Algorithm 1 are exact.

If all interactions are pairwise and each variable is binary and has exactly $|\partial i| = d$ neighbours, the time complexity of the resulting "Loop Corrected BP" (LCBP) algorithm is given by $N 2^d T_{BP} + N d 2^{d+1} N_{LC}$, where T_{BP} is the average time of one BP run on a clamped cavity network and N_{LC} is the number of iterations needed to obtain convergence in Algorithm 1.

2.6 Differences with Montanari and Rizzo (2005)

As mentioned before, the idea of estinating the cavity distributions and imposing certain consistency relations amongst them has been first presented in Montanari and Rizzo (2005). In its simplest form (i.e. the so-called first order correction), the implementation of that basic idea as proposed by Montanari and Rizzo (2005) differs from our proposed implementation in the following aspects.

First, the original method described by Montanari and Rizzo (2005) is only formulated for the rather special case of binary variables and pairwise interactions. In contrast, our method is formulated in a general way that makes it applicable to factor graphs with variables having more than two possible values and factors consisting of more than two variables. Also, factors may contain zeroes. The generality that our implementation offers is important for many practical applications.[3] In the rest of this section, we will assume that the graphical model (1) belongs to the special class of binary variables with pairwise interactions, allowing further comparison of both implementations.

An important difference is that Montanari and Rizzo (2005) suggest to deform the initial approximate cavity distributions by altering certain *cumulants* (also called "connected correlations"), instead of altering certain interactions. In general, for a set $\mathcal{A}$ of ± 1-valued random variables $\{x_i\}_{i\in\mathcal{A}}$, one can define for any subset $\mathcal{B} \subseteq \mathcal{A}$ the *moment*

$$M_{\mathcal{B}} := \sum_{x_{\mathcal{A}}} P(x_{\mathcal{A}}) \prod_{j\in\mathcal{B}} x_j.$$

The moments $\{M_{\mathcal{B}}\}_{\mathcal{B}\subseteq\mathcal{A}}$ are a parameterization of the probability distribution $P(x_{\mathcal{A}})$. An alternative parameterization is given in terms of the cumulants. The *(joint) cumulants* $\{C_{\mathcal{E}}\}_{\mathcal{E}\subseteq\mathcal{A}}$ are certain polynomials of the moments, defined implicitly by the following relations:

$$M_{\mathcal{B}} = \sum_{\mathcal{C}\in\mathrm{Part}(\mathcal{B})} \prod_{\mathcal{E}\in\mathcal{C}} C_{\mathcal{E}}$$

where $\mathrm{Part}(\mathcal{B})$ is the set of partitions of $\mathcal{B}$.[4] In particular, $C_i = M_i$ and $C_{ij} = M_{ij} - M_i M_j$ for all $i, j \in \mathcal{A}$ with $i \neq j$. Montanari and Rizzo (2005) propose to approximate the cavity distributions by estimating the pair cumulants and assuming higher order cumulants to be zero. Then, the singleton cumulants (i.e. the single node marginals) are altered, keeping higher order cumulants fixed, in such a way as to impose consistency of the single node marginals, in the absence of interactions shared by two neighbouring cavities. We refer the reader to the appendix for a more detailed description of the implementation in terms of cumulants suggested by Montanari and Rizzo (2005).

A minor difference lies in the method to obtain initial approximations to the cavity distributions. Montanari and Rizzo (2005) propose to use BP in combination with linear response theory to obtain the initial pairwise cumulants. This difference is not very important, since one could also use BP on clamped cavity networks instead, which turns out to give almost identical results.

3. The method by Montanari and Rizzo (2005) could probably be generalized in a way that stays closer to the original one than our proposal, but it is not so obvious how to do this.
4. For a set X, a *partition* of X is a nonempty set Y such that each $Z \in Y$ is a nonempty subset of X and $\bigcup Y = X$.

As we will show in section 3, our method of altering interactions appears to be more robust and still works in regimes with strong interactions, whereas the cumulant implementation suffers from convergence problems for strong interactions.

An advantage of the cumulant based scheme is that it allows for a linearized version (by expanding up to first order in terms of the pairwise cumulants, see appendix) which is quadratic in the size of the cavity. This means that this linearized, cumulant based version is currently the only one that can be applied to networks with large Markov blankets (cavities), i.e. where the maximum number of states $\max_{i\in\mathcal{V}} |\mathcal{X}_{\Delta i}|$ is large (provided that all variables are binary and interactions are pairwise).

3. Numerical experiments

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

MF Mean Field, with a random sequential update scheme and no damping.

BP Belief Propagation. We have used the recently proposed update scheme Elidan et al. (2006), which converges also for difficult problems without the need for damping.

TreeEP TreeEP (Minka and Qi, 2004), without damping. We generalized the method of choosing the base tree described in Minka and Qi (2004) to multiple variable factors as follows: when estimating the mutual information between x_i and x_j, we take the product of the marginals on $\{i, j\}$ of all the factors that involve x_i and/or x_j. Other generalizations of TreeEP to higher order factors are possible (e.g. by clustering variables), but it is not clear how to do this in general in an optimal way.

LCBP ("Loop Corrected Belief Propagation") Algorithm 1, where the approximate cavities are initialized according to the description in section 2.5.

LCBP-Cum The original cumulant based loop correction scheme by Montanari and Rizzo (2005), using response propagation (also known as linear response; see (Welling and Teh, 2004)) to approximate the initial pairwise cavity cumulants. The full update equations (18) are used and higher order cumulants are assumed to vanish.

LCBP-Cum-Lin Similar to LCBP-Cum, but instead of the full update equations (18), the linearized update equations (19) are used.

CVM-Min A double-loop implementation (Heskes et al., 2003) of the minimal CVM approximation, which uses (maximal) factors as outer clusters.

CVM-Δ A double-loop implementation of CVM using the sets $\{\Delta i\}_{i\in\mathcal{V}}$ as outer clusters. These are the same sets of variables as used by LCBP (c.f. (8)) and therefore it is interesting to compare both algorithms.

CVM-Loop$k$ A double-loop implementation of CVM, using as outer clusters all (maximal) factors together with all loops in the factor graph that consist of up to k different variables (for $k = 3, 4, 5, 6, 8$).

We have used a double-loop implementation of CVM instead of GBP because the former is guaranteed to converge to a local minimum of the Kikuchi free energy (Heskes et al., 2003) without damping, whereas the latter often only converges with strong damping. The difficulty with damping is that the optimal damping constant is not known *a priori*, which necessitates multiple trial runs with different damping constants, until a suitable one is found. Using too much damping slows down convergence, whereas a certain amount of damping is required to obtain convergence in the first place. Therefore, in general we expect that GBP is not much faster than a double-loop implementation because of the computational cost of finding the optimal damping constant.

To be able to assess the errors of the various approximate methods, we have only considered problems for which exact inference (using a standard JunctionTree method) was still feasible.

For each approximate inference method, we report the maximum ℓ_∞ error of the approximate single node marginals b_i, calculated as follows:

$$\text{Error} := \max_{i \in \mathcal{V}} \max_{x_i \in \mathcal{X}_i} |b_i(x_i) - P(x_i)| \tag{12}$$

where $P(x_i)$ is the exact marginal calculated using the JunctionTree method.

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 because we have not spent equal amounts of effort optimizing each method.[5]

We consider an iterative method to be "converged" after T timesteps if for each variable $i \in \mathcal{V}$, the ℓ_∞ distance between the approximate probability distributions of that variable at timestep T and $T+1$ is less than $\epsilon = 10^{-9}$.

We have studied four different model classes: (i) random graphs of uniform degree with pairwise interactions and binary variables; (ii) random factor graphs with binary variables and factor nodes of uniform degree $k = 3$; (iii) the ALARM network, which has variables taking on more than two possible values and factors consisting of more than two variables; (iv) PROMEDAS networks, which have binary variables but factors consisting of more than two variables.

3.1 Random regular graphs with binary variables

We have compared various approximate inference methods on random graphs, consisting of N binary (± 1-valued) variables, having only pairwise interactions, where each variable has the same degree $|\partial i| = d$. In this case, the probability distribution (1) can be written in the following way:

$$P(x) = \frac{1}{Z} \exp\left(\sum_{i \in \mathcal{V}} \theta_i x_i + \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \partial i} J_{ij} x_i x_j \right),$$

where the parameters $\{\theta_i\}_{i \in \mathcal{V}}$ are called the *local fields* and the parameters $\{J_{ij} = J_{ji}\}_{i \in \mathcal{V}, j \in \partial i}$ are called the *couplings*. The graph structure and the parameters θ and J were drawn randomly for each instance.

5. Our C++ implementation of various approximate inference algorithms is free/open source software and can be downloaded from `http://www.mbfys.ru.nl/~jorism/libDAI`

The local fields $\{\theta_i\}$ were drawn independently from a $\mathcal{N}(0, \beta\Theta)$ distribution (i.e. a normal distribution with mean 0 and standard deviation $\beta\Theta$). For the couplings $\{J_{ij}\}$, we distinguished two different cases: mixed ("spin-glass") and attractive ("ferromagnetic") couplings. The couplings were drawn independently from the following distributions:

$$J_{ij} \sim \mathcal{N}\left(0, \beta \tanh^{-1} \frac{1}{\sqrt{d-1}}\right) \qquad \text{mixed couplings}$$

$$J_{ij} = \left|J'_{ij}\right|, \quad J'_{ij} \sim \mathcal{N}\left(0, \beta \tanh^{-1} \frac{1}{d-1}\right) \qquad \text{attractive couplings}$$

The constant β (called "inverse temperature" in statistical physics) controls the overall interaction strength and thereby the difficulty of the inference problem, larger β corresponding usually to more difficult problems. The constant Θ controls the relative strength of the local fields, where larger Θ result in easier inference problems. The particular d-dependent scaling of the couplings is used in order to obtain roughly d-independent behaviour. In case of mixed couplings, for $\Theta = 0$ and for $\beta \approx 1$ a phase transition occurs in the limit of $N \to \infty$, going from an easy "paramagnetic" phase for $\beta < 1$ to a complicated "spin-glass" phase for $\beta > 1$. In the case of attractive couplings and $\Theta = 0$, a phase transition also occurs at $\beta = 1$, now going from the easy "paramagnetic" phase for $\beta < 1$ to a "ferromagnetic" phase for $\beta > 1$.[6]

3.1.1 $N = 100$, $d = 3$, mixed couplings, strong local fields ($\Theta = 2$)

In this section we study regular random graphs of low degree $d = 3$, consisting of $N = 100$ variables, with mixed couplings and relatively strong local fields of strength $\Theta = 2$. We considered various overall interaction strengths β between 0.01 and 10. For each value of β, we used 16 random instances. On each instance, we ran various approximate inference algorithms. Figures 3 and 4 show selected results.[7]

Both figures consist of two parts; the first row in both figures shows averages (in the logarithmic domain) of errors and computation time as a function of β for various methods. In addition, Figure 3 shows the fraction of instances on which each method converged; for Figure 4, all methods converged for all values of β. The averages of errors and computation time were calculated from the converged instances only. The other rows in the figures contain scatter plots that compare errors of various methods one-to-one. The solid red lines in the scatter plots indicate equality; the dotted red lines indicate that the error of the method on the vertical axis is the square of the error on the horizontal axis. Saturation of errors around 10^{-9} is an artefact due to the convergence criterion. The CVM methods are often seen to saturate around 10^{-8}, which indicates that single iterations are less effective than for other methods.

We conclude from both figures that BP is the fastest but also the least accurate method and that LCBP is the most accurate method and that it converges for all β. Furthermore, the error of LCBP is approximately the square of the BP error.

6. More precisely, in case of zero local fields ($\Theta = 0$), the PA-SG phase transition occurs at $(d-1) = \left\langle \tanh^2(\beta J_{ij}) \right\rangle$, where $\langle\cdot\rangle$ is the average over all J_{ij}, and the PA-FE phase transition occurs at $(d-1) = \left\langle \tanh(\beta J_{ij}) \right\rangle$ (Mooij and Kappen, 2005). What happens for $\Theta > 0$ is not known, to the best of our knowledge.
7. We apologize to readers for the use of colours; we saw no viable alternative for creating clear plots.

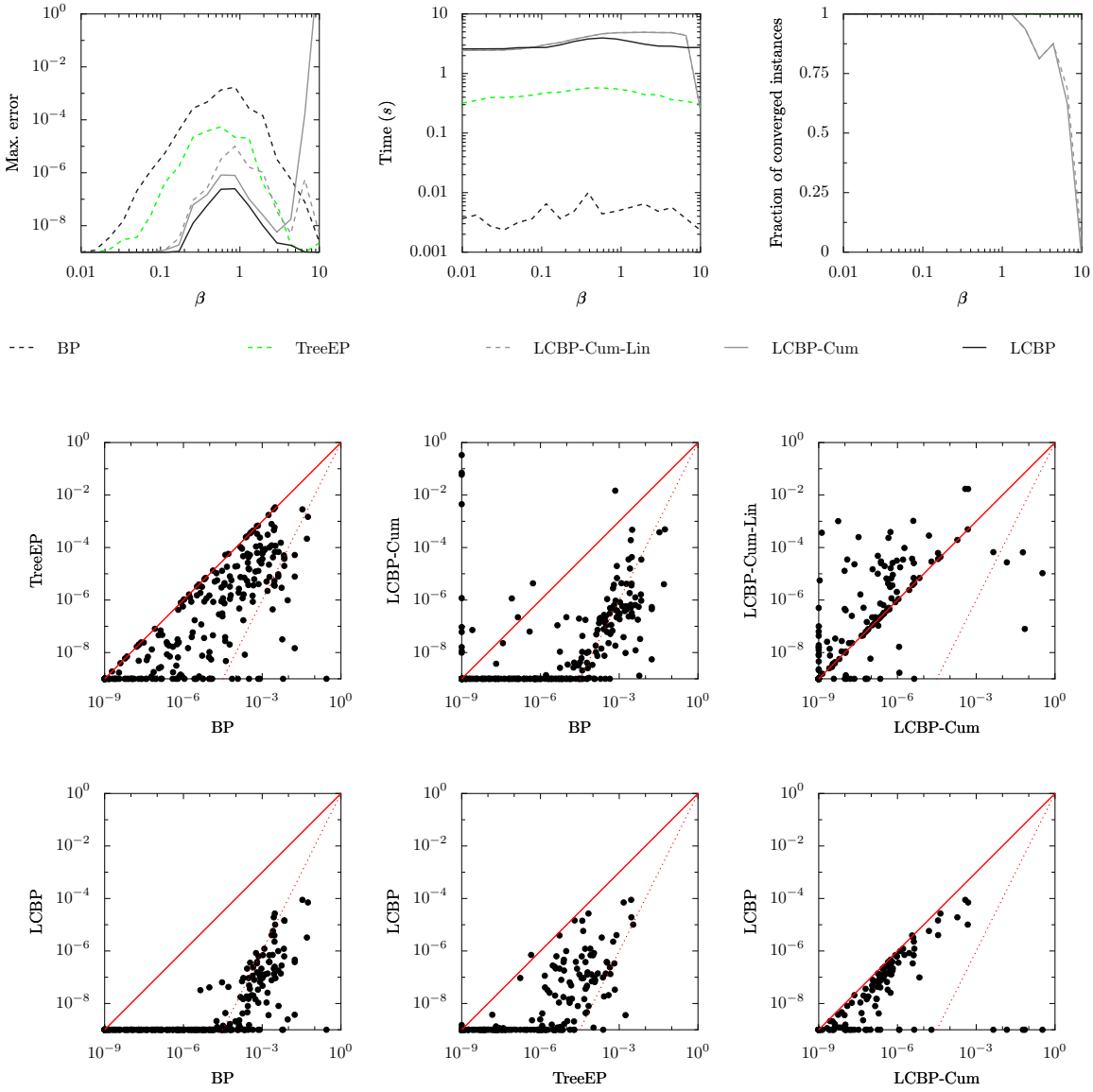


Figure 3: Results for $(N = 100, d = 3)$ regular random graphs with mixed couplings and strong local fields $\Theta = 2$. First row, from left to right: error, computation time and fraction of converged instances, as a function of β for various methods, averaged over 16 randomly generated instances (where results are only included if the method has converged). For the same instances, scatter plots of errors are shown in the next rows for various pairs of methods. The solid red lines correspond with $y = x$, the dotted red lines with $y = x^2$. Only points have been plotted for which both approximate inference methods converged.

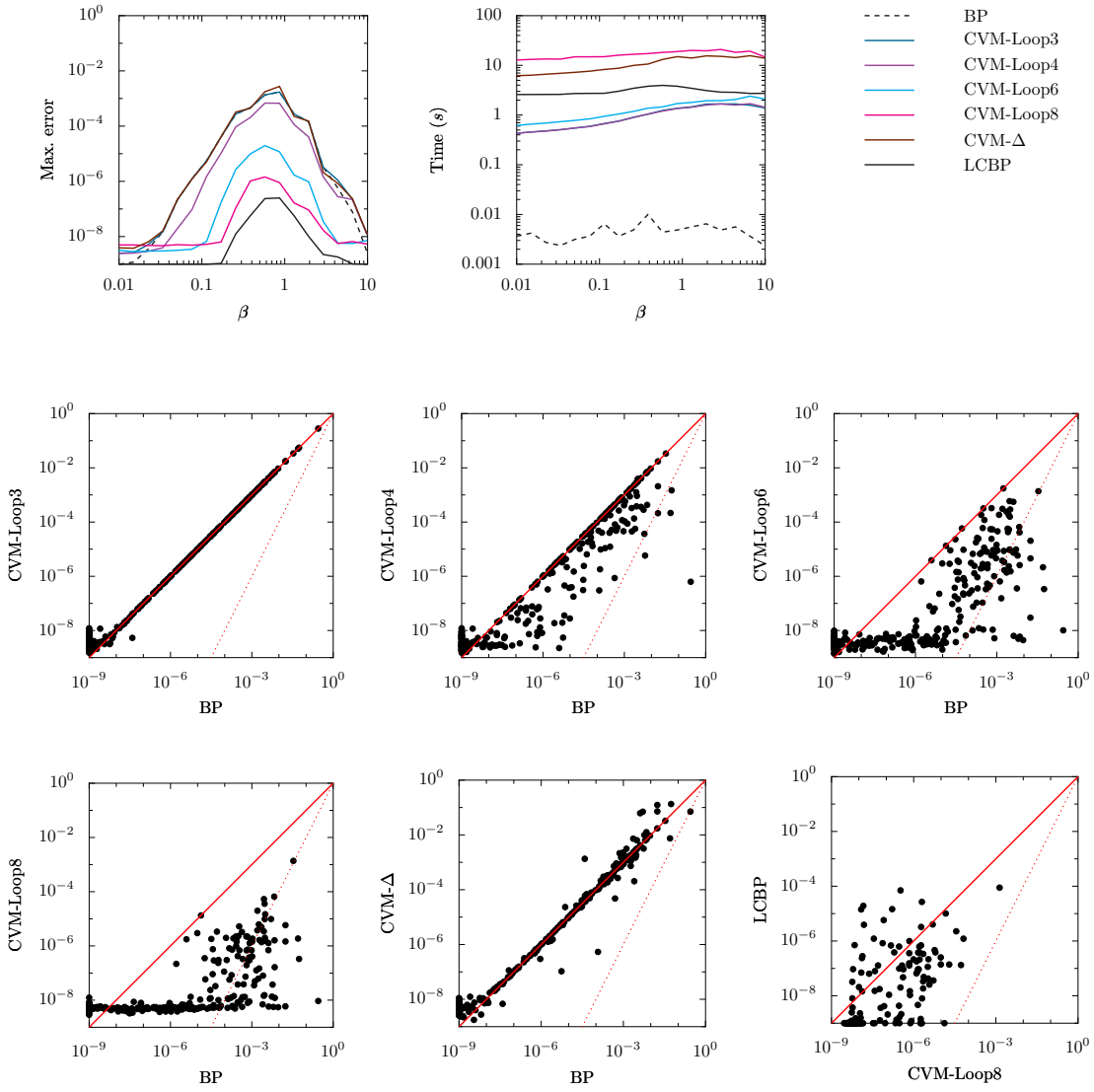


Figure 4: Additional results for $(N = 100, d = 3)$ regular random graphs with mixed couplings and strong local fields $\Theta = 2$, for the same instances as in Figure 3. All methods converged on all instances.

Figure 3 shows further that TreeEP is able to obtain a significant improvement over BP using little computation time. For small values of β, LCBP-Cum and LCBP-Cum-Lin both converge and yield high quality results and the error introduced by the linearization is relatively small. However, for larger values of β, both methods get more and more convergence problems, although for the few cases where they do converge, they still yield accurate results. At $\beta \approx 10$, both methods have completely stopped converging. The error introduced by the linearization increases for larger values of β. The computation times of LCBP-Cum, LCBP-Cum-Lin and LCBP do not differ substantially in the regime where all methods converge. The difference in quality between LCBP and LCBP-Cum is mainly due to the fact that LCBP does take into account triple interactions in the cavity (however, extending LCBP-Cum in order to take into account triple interactions is easy for this case of low d).

The break-down of the cumulant based LCBP methods for high β is probably due to the choice of cumulants for parameterizing cavity distributions, which seem to be less robust than interactions. Indeed, consider two random variables x_1 and x_2 with fixed pair interaction $\exp(Jx_1x_2)$. By altering the singleton interactions $\exp(\theta_1 x_1)$ and $\exp(\theta_2 x_2)$, one can obtain any desired marginals of x_1 and x_2. However, a fixed pair cumulant $C_{12} = \langle x_1 x_2 \rangle - \langle x_1 \rangle \langle x_2 \rangle$ imposes a constraint on the range of possible expectation values $\langle x_1 \rangle$ and $\langle x_2 \rangle$ (hence on the single node marginals of x_1 and x_2); the freedom of choice in these marginals becomes less as the pair cumulant becomes stronger. We believe that something similar happens for LCBP-Cum: for strong interactions, the approximate pair cumulants in the cavity are strong, and even tiny errors can lead to inconsistencies.[8]

The results of the CVM approach to loop correcting is shown in 4. The CVM-Loop methods, with clusters reflecting the short loops present in the factor graph, do improve on BP. The use of larger clusters that subsume longer loops improves the results, but computation time quickly increases. CVM-Loop3 does not obtain any improvement over BP, simply because there are (almost) no loops of 3 variables present. The most accurate CVM method, CVM-Loop8, needs more computation time than LCBP, whereas the quality of its results is not as good. Surprisingly, although CVM-Δ uses larger cluster than BP, its quality is similar to that of BP and its computation time is enormous. This is remarkable, since one would expect that CVM-Δ should improve on BP because it uses larger clusters. In any case, we conclude that although LCBP and CVM-Δ use identical clusters, the nature of both approximations is very different.

We have also done experiments for weak local fields ($\Theta = 0.2$). The behaviour is similar to that of strong local fields, apart from the following differences. First, the influence of the phase transition is more pronounced; many methods have severe convergence problems around $\beta = 1$. Further, the negative effect of linearization on the error (LCBP-Cum-Lin compared to LCBP-Cum) is smaller.

8. Indeed, for strong interactions, the update equations (18) often yield values for the $\mathcal{M}_j^{\setminus i}$ outside of the valid interval $[-1, 1]$. In this case, we project these values back into the valid interval in the hope that the method will converge to a valid result, which it sometimes does. This phenomenon also indicates the lack of robustness of a cumulant parameterization in the regime of strong interactions.

3.1.2 Fixed β and varying relative local field strength Θ

In addition, we have done experiments for fixed $\beta = 1.0$ for various values of the relative local field strength Θ between 0.01 and 10. The results are shown in Figures 5 and 6.

Computation time is seen to decrease with increasing local field strength Θ. The errors on the other hand first increase slowly, and then suddenly decrease rapidly. Again, LCBP-Cum and LCBP-Cum-Lin are the only methods that have convergence problems. The ranking in terms of accuracy of various methods does not depend on the local field strength, nor does the ranking in terms of computation time.

3.1.3 Larger degree ($d = 6$)

To study the influence of the degree $d = |\partial i|$, we have done additional experiments for $d = 6$. We had to reduce the number of variables to $N = 50$, because exact inference was infeasible for larger values of N due to quickly increasing treewidth. The results are shown in Figure 7.

As in the previous experiments, BP is the fastest and least accurate method, whereas LCBP yields the most accurate results, even for high β.

The differences with the case of low degree ($d = 3$) are the following. The relative improvement of TreeEP over BP has decreased. This could have been expected, because in denser networks, the effect of taking out a tree becomes less. Further, the relative improvement of CVM-Loop4 over BP has increased, probably because there are more short loops present. On the other hand, computation time of CVM-Loop4 has also increased and it is the slowest of all methods. We decided to abort the calculations for CVM-Loop6 and CVM-Loop8, because computation time was prohibitive due to the enormous amount of short loops present. We conclude that the CVM-Loop approach to loop correcting is not very efficient. Surprisingly, the results of LCBP-Cum-Lin are now very similar in quality to the results of LCBP-Cum, except for a few isolated cases (presumably on the edge of the convergence region). LCBP now clearly needs more computation time than LCBP-Cum and LCBP-Cum-Lin, but also obtains significantly better results due to the fact that it takes into account higher order cavity interactions.

3.1.4 Influence of the coupling type

To study the influence of coupling type, we have done additional experiments for ($N = 50, d = 6$) random regular graphs with attractive couplings and strong local fields ($\Theta = 2$). The results are shown in 8.

By comparing Figures 7 and 8, it becomes clear that the influence of the coupling type is rather small. Differences might be more pronounced in case of weak local fields (for which we have not done additional experiments).

3.1.5 Scaling with N

We have investigated how computation time scales with the number of variables N, for fixed $\beta = 0.1$, $\Theta = 2$ and $d = 6$ for mixed couplings. We used a machine with more memory (16 GB) to be able to do exact inference without swapping also for $N = 60$. The results can be

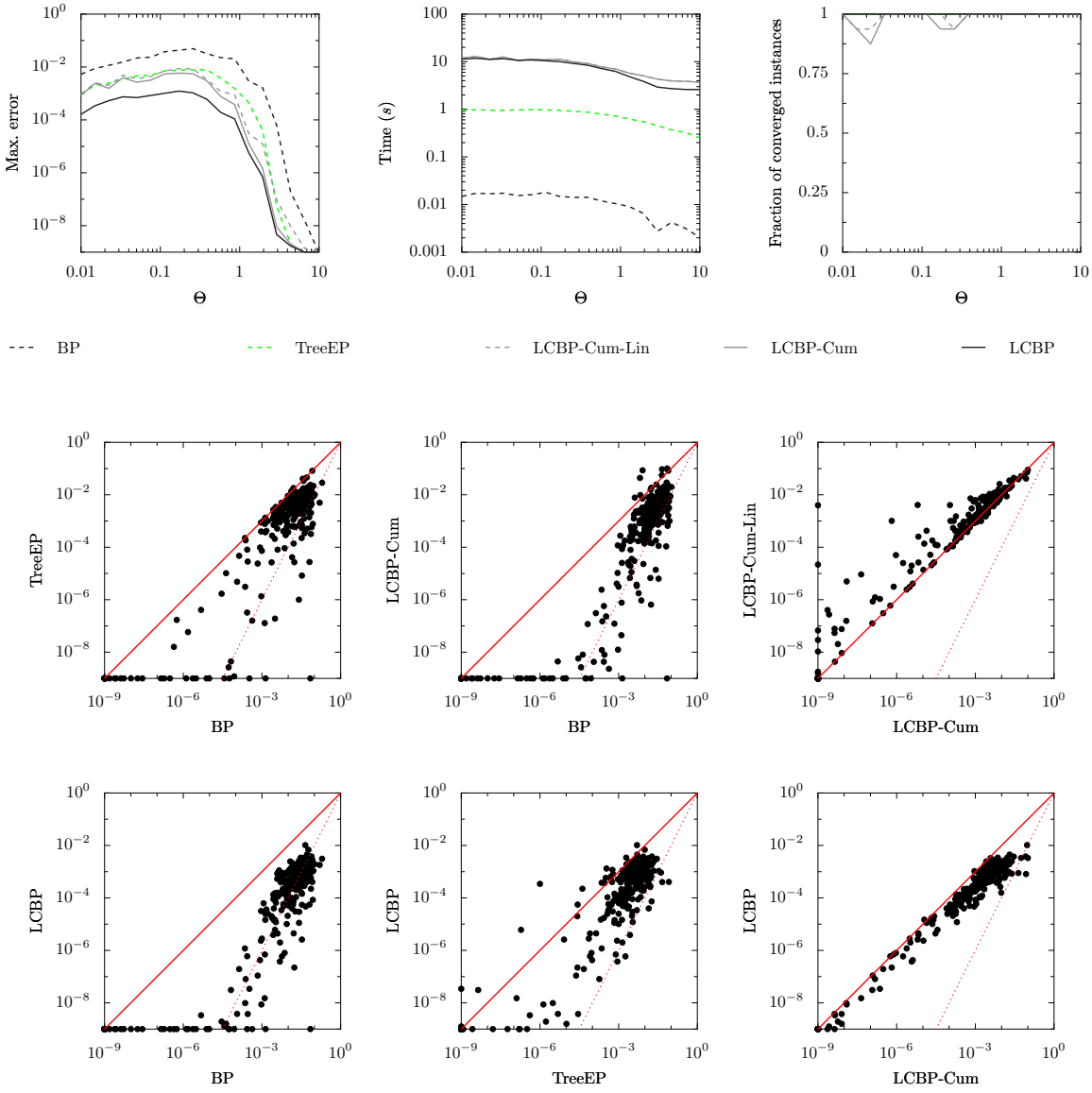


Figure 5: Results for $(N = 100, d = 3)$ regular random graphs with mixed couplings and $\beta = 1.0$. First row, from left to right: error, computation time and fraction of converged instances, as a function of relative local field strength Θ for various methods, averaged over 16 randomly generated instances (where results are only included if the method has converged). For the same instances, scatter plots of errors are shown in the next rows for various pairs of methods.

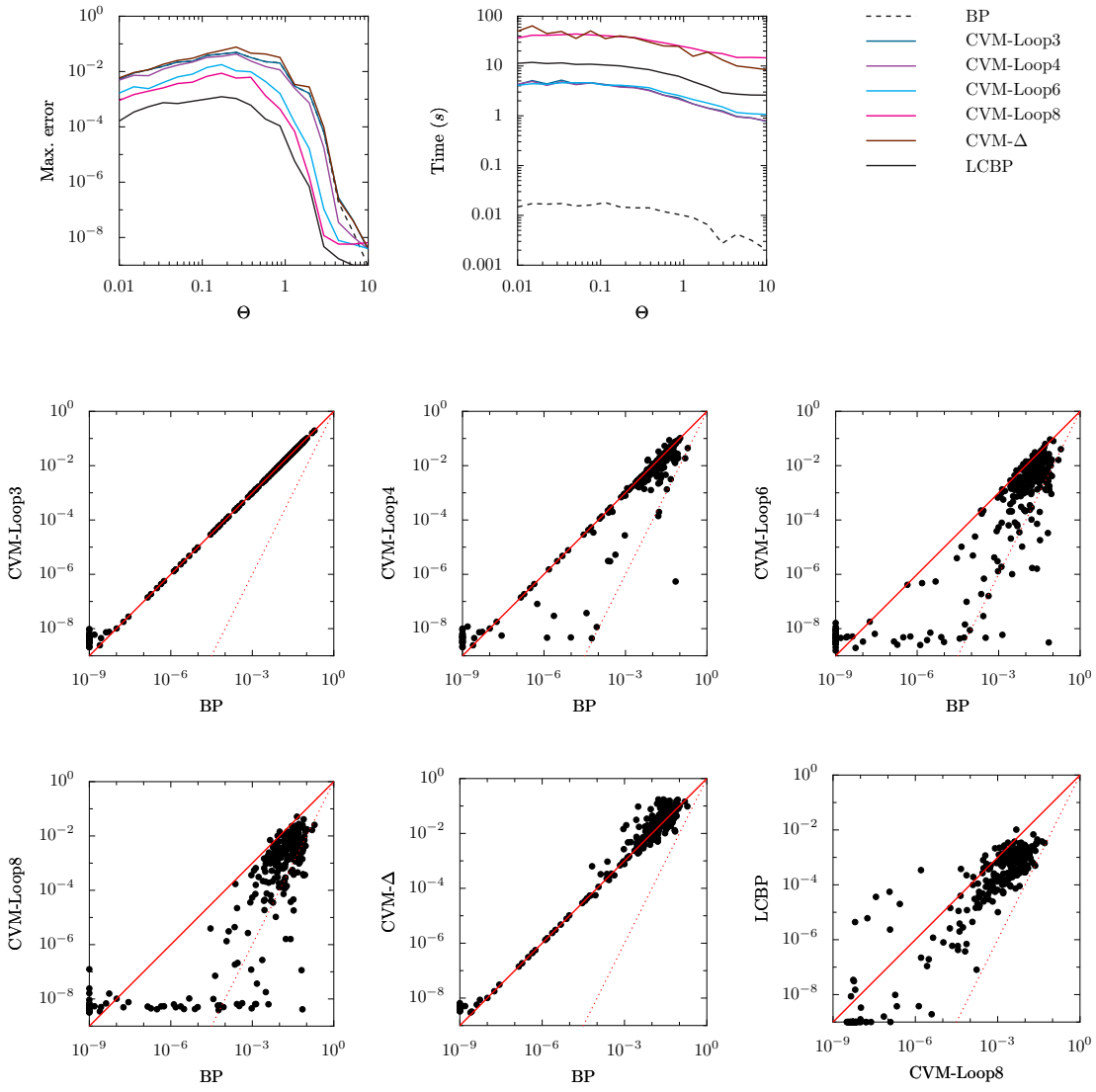


Figure 6: Additional results for $(N = 100, d = 3)$ regular random graphs with mixed couplings and $\beta = 1.0$, for the same instances as in Figure 5. All methods converged on all instances.

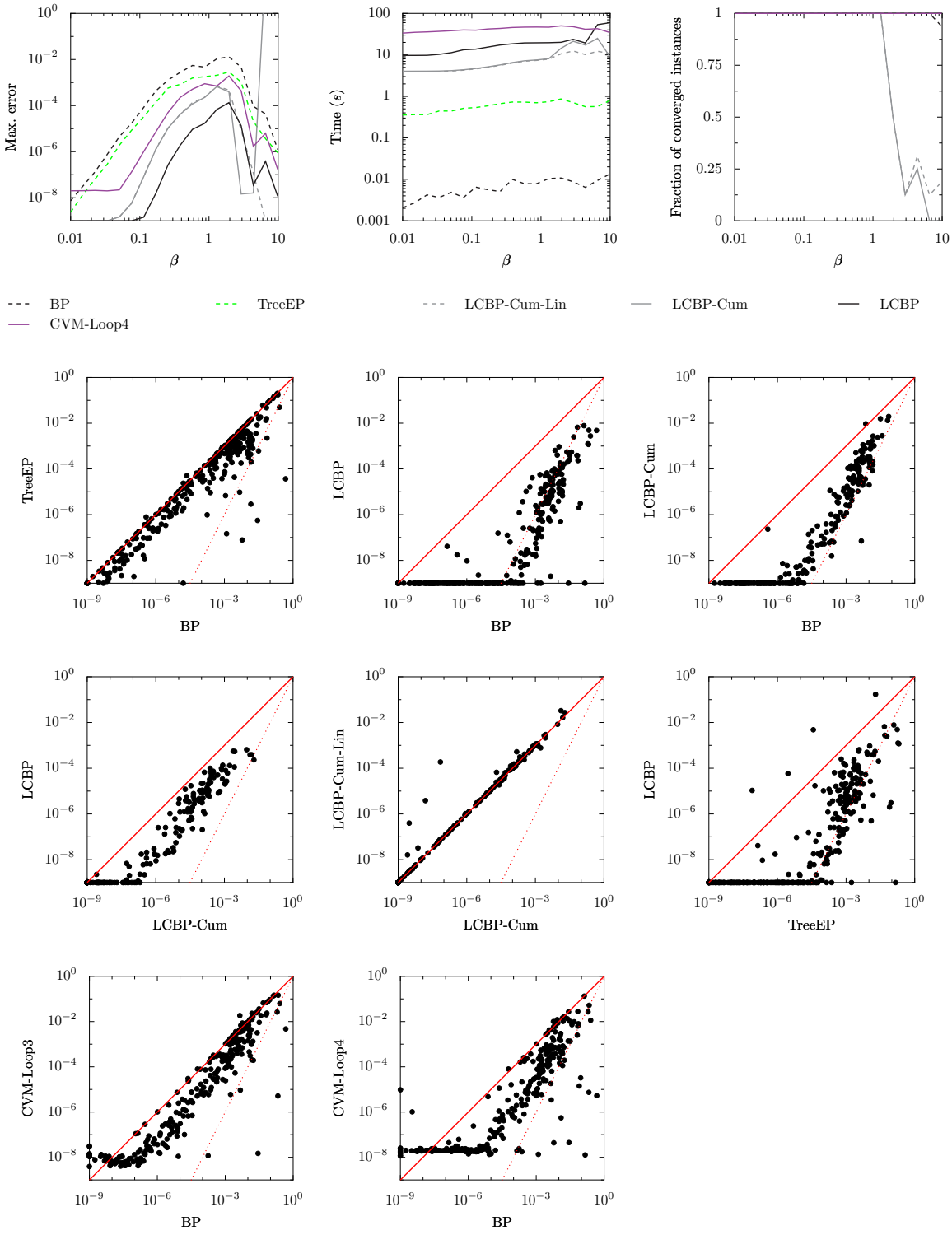


Figure 7: Results for $(N = 50, d = 6)$ regular random graphs with mixed couplings and strong local fields $\Theta = 2$.

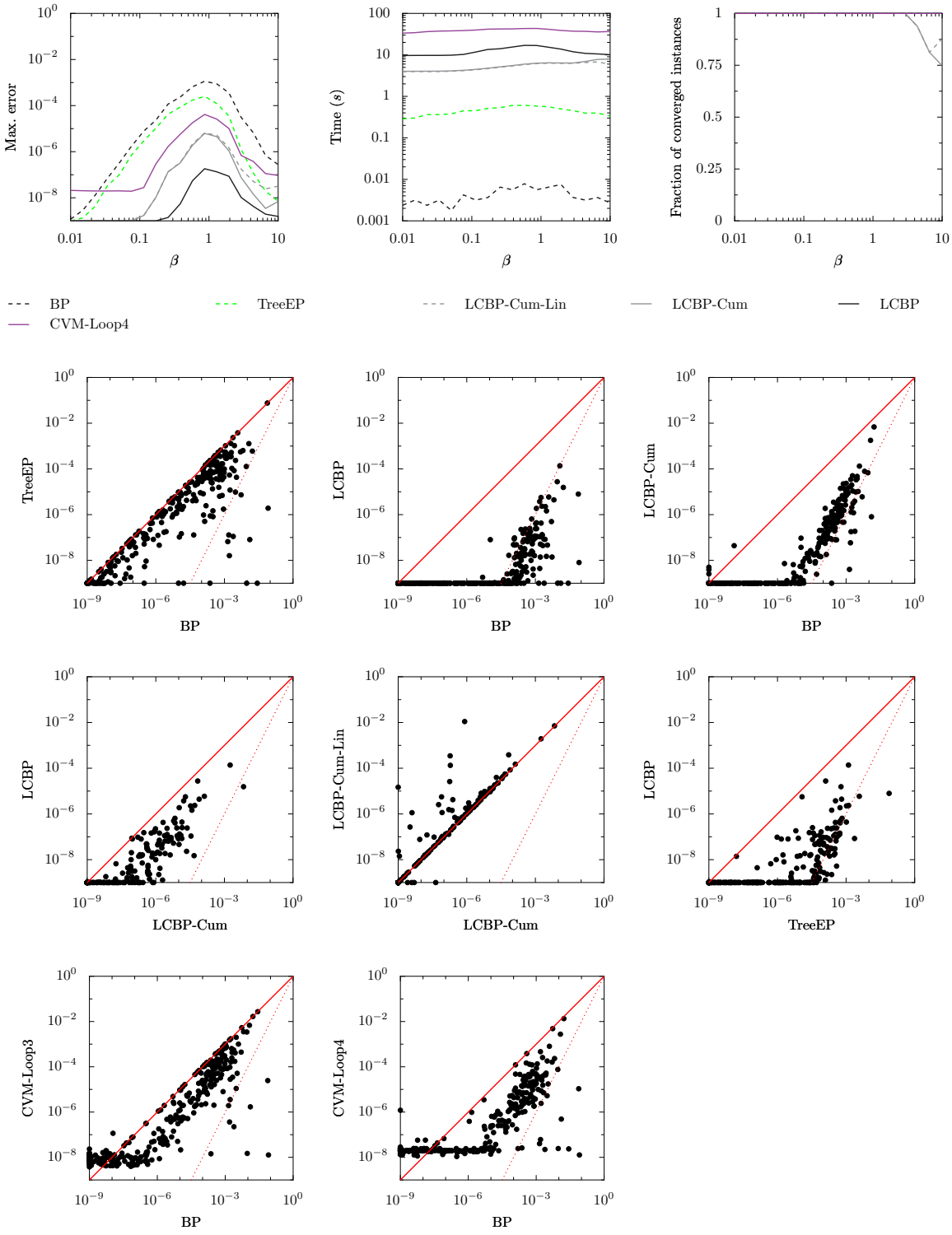


Figure 8: Results for $(N = 50, d = 6)$ regular random graphs with attractive couplings and strong local fields $\Theta = 2$.

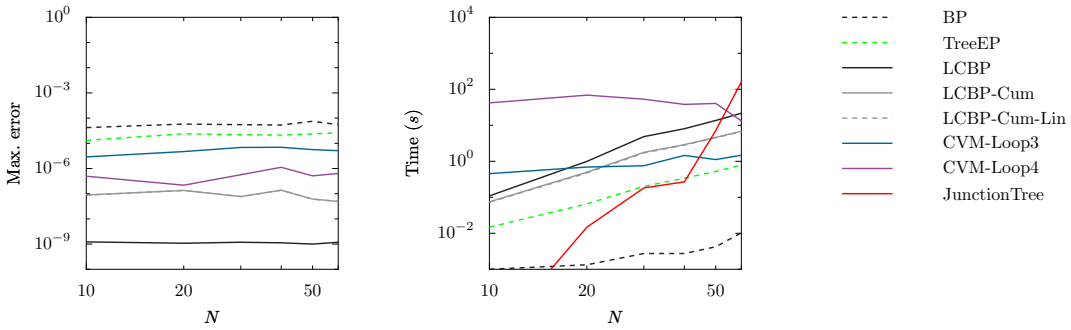


Figure 9: Error (left) and computation time (right) as a function of N (the number of variables), for random graphs with uniform degree $d = 6$, mixed couplings, $\beta = 0.1$ and $\Theta = 2$. Points are averages over 16 randomly generated instances. Each method converged on all instances.

found in Figure 9. For larger values of N, the computation time for exact inference would increase exponentially with N.

The error of all methods is approximately constant. BP should scale approximately linearly in N. LCBP variants are expected to scale quadratic in N (since d is fixed) which indeed appears to be the case. The computation time of the exact JunctionTree method quickly increases due to increasing treewidth; for $N = 60$ it is already ten times larger than the computation time of the slowest approximate inference method. The computation time of CVM-Loop3 and CVM-Loop4 seems to be approximately constant, probably because the large number of overlaps of short loops for small values of N causes difficulties.

We conclude that for large N, exact inference is infeasible, whereas LCBP still yields very accurate results using moderate computation time.

3.2 Scaling with d

It is also interesting to see how various methods scale with d, the variable degree, which is directly related to the cavity size. We have done experiments for random graphs of size $N = 24$ with fixed $\beta = 0.1$ and $\Theta = 2$ for mixed couplings for different values of d between 3 and 23. The results can be found in Figure 10. We aborted the calculations of the slower methods (LCBP, LCBP-Cum, CVM-Loop3) at $d = 15$.

Due to the particular dependence of the interaction strength on d, the errors of most methods depend only slightly on d. TreeEP is an exception: for larger d, the relative improvement of TreeEP over BP diminishes, and the TreeEP error approaches the BP error. CVM-Loop3 gives better quality, but needs relatively much computation time and becomes very slow for large d due to the large increase in the number of loops of 3 variables. LCBP is the most accurate method, but becomes very slow for large d. LCBP-Cum is less accurate and becomes slower than LCBP for large d, because of the additional overhead of the combinatorics needed to perform the update equations. The accuracy of LCBP-Cum-Lin is indistinguishable from that of LCBP-Cum, although it needs significantly less computation time.

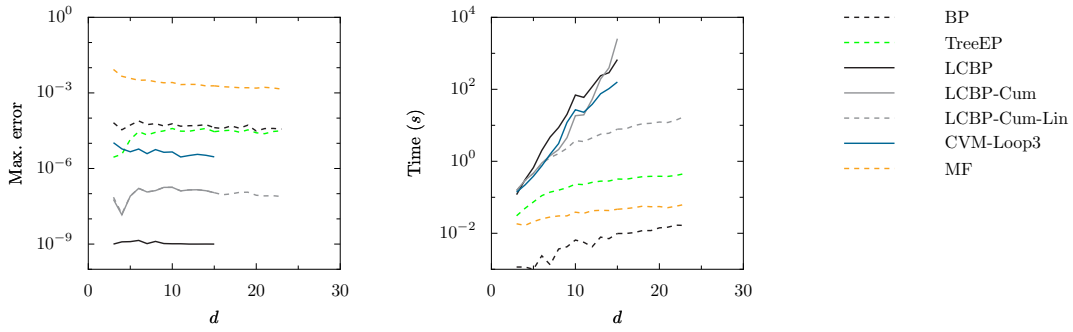


Figure 10: Error (left) and computation time (right) as a function of variable degree d for regular random graphs of $N = 24$ variables with mixed couplings for $\beta = 0.1$ and $\Theta = 2$. Points are averages over 16 randomly generated instances. Each method converged on all instances.

3.3 Alternative methods to obtain initial approximate cavity distributions

Until now we have used BP to estimate initial cavity approximations. We now show that other approximate inference methods can be used as well and that a similar relative improvement in accuracy is obtained. Figure 11 shows the results of Algorithm 1 for cavity approximations initialized using the method described in Section 2.5 with MF and TreeEP instead of BP. For reference, also the BP results are plotted. In all cases, the loop corrected error is approximately the square of the error of the uncorrected approximate inference method. Because BP is very fast yet relatively accurate, we focus on LCBP in this article.

3.4 Multi-variable factors

We now go beyond pairwise interactions and study a class of random factor graphs with binary variables and uniform factor degree $|I| = k$ (for all $I \in \mathcal{F}$) with $k > 2$. The number of variables is N and the number of factors is M. The factor graphs are constructed by starting from an empty graphical model $(\mathcal{V}, \emptyset, \emptyset)$ and adding M random factors, where each factor is obtained in the following way: a subset $I = \{I_1, \ldots, I_k\} \subseteq \mathcal{V}$ of k different variables is drawn; a vector of 2^k independent random numbers $\{J_I(x_I)\}_{x_I \in \mathcal{X}_I}$ is drawn from a $\mathcal{N}(0, \beta)$ distribution; the factor $\psi_I(x_I) := \exp J_I(x_i)$ is added to the graphical model. We only use those constructed factor graphs that are connected.[9] The parameter β again controls the interaction strength.

We have done experiments for $(N = 50, M = 50, k = 3)$ for various values of β between 0.01 and 2. For each value of β, we have used 16 random instances. For higher values of β, computation times increased and convergence became problematic for some methods, which can probably be explained as the effects of a phase transition. The results are shown in Figure 12.

9. The reason that we require the factor graph to be connected is that not all our approximate inference method implementations currently support connected factor graphs that consist of more than one connected component.

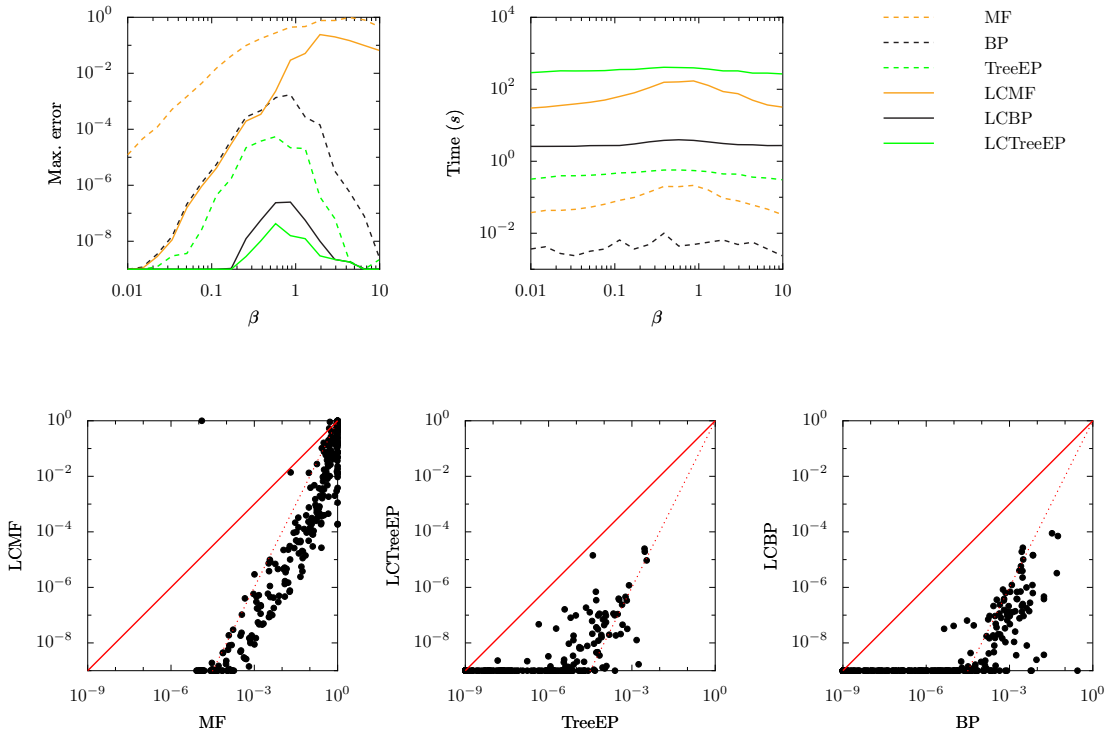


Figure 11: Results for different methods of obtaining initial estimates of cavity distributions, for $(N = 100, d = 3)$ regular random graphs with mixed couplings and strong local fields $\Theta = 2$.

Looking at the error and the computation time in Figure 12, the following ranking can be made, where accuracy and computation time both increase: BP, TreeEP, CVM-Min, CVM-Loop3, LCBP. CVM-Loop4 uses more computation time than LCBP but gives worse results. LCBP-Cum and LCBP-Cum-Lin are not available due to the fact that the factors involve more than two variables. The improvement of TreeEP over BP is rather small.

3.5 ALARM network

The ALARM network[10] is a well-known Bayesian network consisting of 37 variables (some of which can take on more than two possible values) and 37 factors (many of which involve more than two variables). In addition to the usual approximate inference methods, we have compared with GBP-Min, a GBP implementation of the minimal CVM approximation that uses maximal factors as outer clusters. The results are reported in Table 1.

The accuracy of GBP-Min (and CVM-Min) is almost identical to that of BP for this graphical model; GBP-Min converges without damping and is faster than CVM-Min. TreeEP on the other hand significantly improves the BP result in roughly the same time as GBP-Min needs. Simply enlarging the cluster size (CVM-Δ) slightly deteriorates the quality

10. The ALARM network can be downloaded from `http://compbio.cs.huji.ac.il/Repository/Datasets/alarm/alarm.dsc`

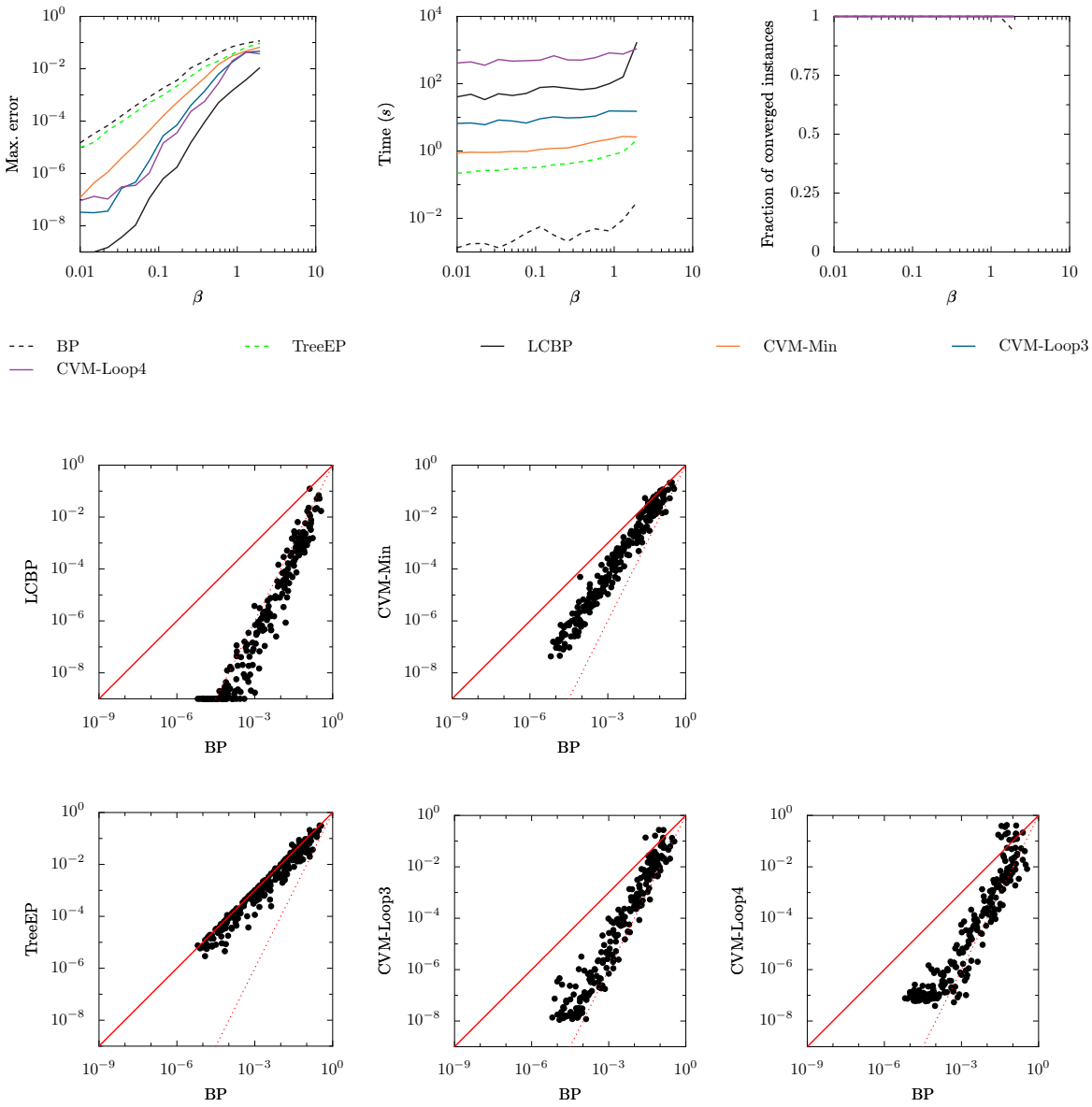


Figure 12: Results for $(N = 50, M = 50, k = 3)$ random factor graphs.

of the results and also causes an enormous increase of computation time. The quality of the CVM-Loop results is roughly comparable to that of TreeEP. Suprisingly, increasing the loop depth beyond 4 deteriorates the quality of the results and results in an explosion of computation time. We conclude that the CVM-Loop method is not a very good approach to correcting loops in this case. LCBP uses considerable computation time, but yields errors that are approximately 10^4 times smaller than BP errors. The cumulant based loop LCBP methods are not available, due to the presence of factors involving more than two variables and variables that can take more than two values.

3.6 PROMEDAS networks

In this subsection, we study the performance of LCBP on another "real-world" example, the PROMEDAS medical diagnostic network (Wiegerinck et al., 1999). The diagnostic model in PROMEDAS is based on a Bayesian network. The global architecture of this network is similar to QMR-DT (Shwe et al., 1991). It consists of a diagnosis-layer that is connected to a layer with findings[11]. Diagnoses (diseases) are modeled as *a priori* independent binary variables causing a set of symptoms (findings), which constitute the bottom layer. The PROMEDAS network currently consists of approximately 2000 diagnoses and 1000 findings.

The interaction between diagnoses and findings is modeled with a noisy-OR structure. The conditional probability of the finding given the parents is modeled by $m+1$ numbers, m of which represent the probabilities that the finding is caused by one of the diseases and one that the finding is not caused by any of the parents.

The noisy-OR conditional probability tables with m parents can be naively stored in a table of size 2^m. This is problematic for the PROMEDAS networks since findings that are affected by more than 30 diseases are not uncommon in the PROMEDAS network. We use an efficient implementation of noisy-OR relations as proposed by Takikawa and D'Ambrosio (1999) to reduce the size of these tables. The trick is to introduce dummy variables s and

11. In addition, there is a layer of variables, such as age and gender, that may affect the prior probabilities of the diagnoses. Since these variables are always clamped for each patient case, they merely change the prior disease probabilities and are irrelevant for our current considerations.

Table 1: Results for the ALARM network

Method	Time (s)	Error
BP	0.00	$2.026 \cdot 10^{-01}$
TreeEP	0.21	$3.931 \cdot 10^{-02}$
GBP-Min	0.18	$2.031 \cdot 10^{-01}$
CVM-Min	1.13	$2.031 \cdot 10^{-01}$
CVM-Δ	280.67	$2.233 \cdot 10^{-01}$
CVM-Loop3	1.19	$4.547 \cdot 10^{-02}$
CVM-Loop4	154.97	$3.515 \cdot 10^{-02}$
CVM-Loop5	1802.83	$5.316 \cdot 10^{-02}$
CVM-Loop6	84912.70	$5.752 \cdot 10^{-02}$
LCBP	23.67	$3.412 \cdot 10^{-05}$

to make use of the property

$$\mathrm{OR}(x|y_1, y_2, y_3) = \sum_s \mathrm{OR}(x|y_1, s)\mathrm{OR}(s|y_2, y_3) \tag{13}$$

The factors on the right hand side involve at most 3 variables instead of the initial 4 (left). Repeated application of this formula reduces all factors to triple interactions or smaller.

When a patient case is presented to PROMEDAS, a subset of the findings will be clamped and the rest will be unclamped. If our goal is to compute the marginal probabilities of the diagnostic variables only, the unclamped findings and the diagnoses that are not related to any of the clamped findings can be summed out of the network as a preprocessing step. The clamped findings cause an effective interaction between their parents. However, the noisy-OR structure is such that when the finding is clamped to a negative value, the effective interaction factorizes over its parents. Thus, findings can be clamped to negative values without additional computation cost (Jaakkola and Jordan, 1999).

The complexity of the problem now depends on the set of findings that is given as input. The more findings are clamped to a positive value, the larger the remaining network of disease variables and the more complex the inference task. Especially in cases where findings share more than one common possible diagnosis, and consequently loops occur, the model can become complex.

We use the PROMEDAS model to generate virtual patient data by first clamping one of the disease variables to be positive and then clamping each finding to its positive value with probability equal to the conditional distribution of the finding, given the positive disease. The union of all positive findings thus obtained constitute one patient case. For each patient case, the corresponding truncated graphical model is generated. The number of disease nodes in this truncated graph is typically quite large.

The results can be found in Figures 13 and 14. Surprisingly, neither TreeEP nor any of the CVM methods gives substantial improvements over BP. TreeEP even gives worse results compared to BP. The CVM-Min and CVM-Loop3 results appear to be almost identical to the BP results. CVM-Loop4 manages to improve over BP in a few cases. Increased loop depth ($k = 5, 6$) results in worse quality in many cases and also in an enormous increase in computation time.

LCBP, on the other hand, is the only method that gives a significant improvement over BP, in each case. Considering all patient cases, LCBP corrects the BP error with more than one order of magnitude in half of the cases for which BP was not already exact. The improvement obtained by LCBP has its price: the computation time of LCBP is rather large compared to that of BP, as shown in Figure 14. The deviation from the quadratic scaling $t \propto N^2$ is due to the fact that the size of the Markov blankets varies over instances and instances with large N often also have larger Markov blankets. The cumulant based loop LCBP methods are not available, due to the presence of factors involving more than two variables and variables that can take more than two values.

4. Discussion and conclusion

We have proposed a method to improve the quality of an approximate inference method (e.g. BP) by correcting for the influence of loops in the factor graph. We found empirically that

if one applies this Loop Correcting method, assuming that no loops are present (by taking factorized initial approximate cavity distributions), the method reduces to the minimal CVM approximation. We have proved this for the case of factor graphs that do not have short loops of exactly four nodes. If, on the other hand, the loop correction method is applied in combination with BP estimates of the effective cavity interactions, we have seen that the loop corrected error is approximately the square of the uncorrected BP error. Similar observations have been made for loop corrected MF and TreeEP. For practical purposes, we suggest to apply loop corrections to BP ("LCBP"), because the loop correction approach requires many runs of the approximate inference method and BP is well suited for this job because of its speed. We have compared the performance of LCBP with other approximate inference methods that (partially) correct for the presence of loops. We have shown that LCBP is the most accurate method and that it even works for relatively strong interactions.

On sparse factor graphs, TreeEP obtains improvements over BP by correcting for loops that consist of part of the base tree and one additional interaction, using little computation time. However, for denser graphs, the difference between the quality of TreeEP and BP marginals diminishes. LCBP almost always obtained more accurate results. However, LCBP also needs more computation time than TreeEP.

The CVM-Loop approximation, which uses small loops as outer clusters, can also provide accurate results if the number of short loops is not too large and the number of intersections of clusters is limited. However, the computation time becomes prohibitive in many cases. In order to obtain the same accuracy as LCBP, the CVM-Loop approach usually needs significantly more computation time. This behaviour is also seen on "real world" instances such as the ALARM network and PROMEDAS test cases. There may exist other cluster choices that give better results for the CVM approximation, but no general method for obtaining "good" cluster choices seems to be known (see also (Welling et al., 2005) for a discussion of what constitutes a "good" CVM cluster choice).

We have also compared the performance of LCBP with the original implementation proposed by Montanari and Rizzo (2005). This implementation works with cumulants instead of interactions and we believe that this is the reason that it has more difficulties in the regime of strong interactions. Although the differences were rather small in some cases, LCBP obtained better results than LCBP-Cum using approximately similar amounts of computation time. The linearized version LCBP-Cum-Lin, which is applicable to factor graphs with large Markov blankets, performed surprisingly well, often obtaining similar accuracy as LCBP-Cum. For random graphs with high degree d (i.e. large Markov blankets), it turned out to be the most accurate of the applicable approximate inference methods. It is rather fortunate that the negative effect of the linearization error on the accuracy of the result becomes smaller as the degree increases, since it is precisely for high degree where one needs the linearization because of performance issues.

In the experiments reported here, the standard JunctionTree method was almost always faster than LCBP. The reason is that we have intentionally selected experiments for which exact inference is still feasible, in order to be able to compare the quality of various approximate inference methods. However, as implied by Figure 9, there is no reason to expect that LCBP will suddenly give inaccurate results when exact inference is no longer feasible. Thus we suggest that LCBP may be used to obtain accurate marginal estimates in cases where exact inference is impossible because of high treewidth. As illustrated in

Figure 9, the computation time of LCBP scales very different from that of the JunctionTree method: whereas the latter is exponential in treewidth, LCBP is exponential in the size of the Markov blankets.

The fact that computation time of LCBP (in its current form) scales exponentially with the size of the Markov blankets can be a severe limitation in practice. Many real world Bayesian networks have large Markov blankets, prohibiting application of LCBP. The linear cumulant based implementation LCBP-Cum-Lin proposed by Montanari and Rizzo (2005) does not suffer from this problem, as it is quadratic in the size of the Markov blankets. Unfortunately, this particular implementation can in its current form only be applied to graphical models that consist of binary variables and factors that involve at most two variables (which excludes any interesting Bayesian network, for example). Furthermore, problems may arise if some factors contain zeroes. For general application of loop correcting methods, it will be of paramount importance to derive an implementation that combines the generality of LCBP with the speed of LCBP-Cum-Lin. At this point, it is not obvious whether it would be better to use cumulants or interactions as the parameterization of the cavity distribution. This topic will be left for future research. The work presented here provides some intuition that may be helpful for constructing a general and fast loop correcting method that is applicable to arbitrary factor graphs that can have large Markov blankets.

Another important direction for future research would be to find an extension of the loop correcting framework that also gives a loop corrected approximation of the normalization constant Z in (1). Additionally, (and possibly related to that), it would be desirable to find an approximate "free energy", a function of the beliefs, whose stationary points coincide with the fixed points of the Algorithm 1. This can be done for many approximate inference methods (MF, BP, CVM, EP) so it is natural to expect that the Loop Correction algorithm can also be seen as a minimization procedure of a certain approximate free energy. Despite some efforts, we have not yet been able to find such a free energy.

Recently, other loop correcting approaches (to the Bethe approximation) have been proposed in the statistical physics community (Parisi and Slanina, 2005; Chertkov and Chernyak, 2006). In particular, Chertkov and Chernyak (2006) have derived a series expansion of the *exact* normalizing constant Z in terms of the BP solution. The first term of the series is precisely the Bethe free energy evaluated at the BP fixed point. The number of terms in the series is finite, but can be very large, even larger than the number of total states of the graphical model. Each term is associated with a "generalized loop", which is a subgraph of the factor graph for which each node has at least connectivity two. By truncating the series, it is possible to obtain approximate solutions that improve on BP by taking into account a subset of all generalized loops (Gómez et al., 2006). Summarizing, this approach to loop corrections takes a subset of loops into account in an exact way, whereas the loop correcting approach presented in this article takes all loops into account in an approximate way. More experiments should be done to compare both approaches.

Concluding, we have proposed a method to correct approximate inference methods for the influence of loops in the factor graph. We have shown that it can obtain very accurate results, also on real world graphical models, outperforming existing approximate inference methods in terms of quality, robustness or applicability. We have shown that it can be applied to problems for which exact inference is infeasible. The rather large computation

time required is an issue which deserves further consideration; it may be possible to use additional approximations on top of the loop correcting framework that trade quality for computation time.

Acknowledgments

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024. We thank Bastian Wemmenhove for stimulating discussions and for providing the PROMEDAS test cases.

Appendix: Original approach proposed by Montanari and Rizzo (2005)

For completeness, we describe the implementation based on cumulants as originally proposed by Montanari and Rizzo (2005). The approach can be applied in recursive fashion. Here we will only discuss the first recursion level.

Consider a graphical model which has only binary (± 1-valued) variables and factors that involve at most two variables. The corresponding probability distribution can be parameterized in terms of the local fields $\{\theta_i\}_{i\in\mathcal{V}}$ and the couplings $\{J_{ij} = J_{ji}\}_{i\in\mathcal{V},j\in\partial i}$:

$$P(x) = \frac{1}{Z}\exp\left(\sum_{i\in\mathcal{V}}\theta_i x_i + \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\partial i} J_{ij}x_i x_j\right).$$

Let $i \in \mathcal{V}$ and consider the corresponding cavity network of i. For $\mathcal{A} \subseteq \partial i$, the cavity *moment* $\mathcal{M}_{\mathcal{A}}^{\backslash i}$ is defined as the following expectation value under the cavity distribution:

$$\mathcal{M}_{\mathcal{A}}^{\backslash i} := \sum_{x_{\partial i}} Z^{\backslash i}(x_{\partial i}) \prod_{j\in\mathcal{A}} x_j,$$

where we will not explicitly distinguish between approximate and exact quantities, following the physicists' tradition.[12] The cavity *cumulants* (also called "connected correlations") $\mathcal{C}_{\mathcal{A}}^{\backslash i}$ are related to the moments in the following way:

$$\mathcal{M}_{\mathcal{A}}^{\backslash i} = \sum_{\mathcal{B}\in\mathrm{Part}(\mathcal{A})}\prod_{\mathcal{E}\in\mathcal{B}}\mathcal{C}_{\mathcal{E}}^{\backslash i}$$

where $\mathrm{Part}(\mathcal{A})$ is the set of partitions of $\mathcal{A}$.

We introduce some notation: we define for $\mathcal{A} \subseteq \partial i$:

$$t_{i\mathcal{A}} := \prod_{k\in\mathcal{A}} \tanh J_{ik}.$$

Further, for a set X, we denote the even subsets of X as $\mathcal{P}_+(X) := \{Y \subseteq X : |Y| \text{ is even}\}$ and the odd subsets of X as $\mathcal{P}_-(X) := \{Y \subseteq X : |Y| \text{ is odd}\}$.

Using standard algebraic manipulations, one can show that for $j \in \partial i$, the expectation value of x_j in the absence of the interaction $\psi_{ij} = \exp(J_{ij}x_i x_j)$ can be expressed in terms of cavity moments of i as follows:

$$\frac{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial i\backslash j)} t_{i\mathcal{A}}\mathcal{M}_{\mathcal{A}\cup j}^{\backslash i} + \tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i\backslash j)} t_{i\mathcal{A}}\mathcal{M}_{\mathcal{A}\cup j}^{\backslash i}}{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial i\backslash j)} t_{i\mathcal{A}}\mathcal{M}_{\mathcal{A}}^{\backslash i} + \tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i\backslash j)} t_{i\mathcal{A}}\mathcal{M}_{\mathcal{A}}^{\backslash i}}. \tag{14}$$

On the other hand, the same expectation value can also be expressed in terms of cavity moments of j as follows:

$$\frac{\tanh\theta_j \sum_{\mathcal{A}\in\mathcal{P}_+(\partial j\backslash i)} t_{j\mathcal{B}}\mathcal{M}_{\mathcal{B}}^{\backslash j} + \sum_{\mathcal{A}\in\mathcal{P}_-(\partial j\backslash i)} t_{j\mathcal{B}}\mathcal{M}_{\mathcal{B}}^{\backslash j}}{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial j\backslash i)} t_{j\mathcal{B}}\mathcal{M}_{\mathcal{B}}^{\backslash j} + \tanh\theta_j \sum_{\mathcal{A}\in\mathcal{P}_-(\partial j\backslash i)} t_{j\mathcal{B}}\mathcal{M}_{\mathcal{B}}^{\backslash j}}. \tag{15}$$

12. In (Montanari and Rizzo, 2005), the notation $\tilde{C}_{\mathcal{A}}^{(i)}$ is used for the cavity moment $\mathcal{M}_{\mathcal{A}}^{\backslash i}$.

The consistency equations are now given by equating (14) to (15) for all $i \in \mathcal{V}$, $j \in \partial i$.

The expectation value of x_i (in the presence of all interactions) can be similarly expressed in terms of cavity moments of i:

$$M_i := \sum_{x_i=\pm 1} P(x_i) x_i = \frac{\tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_+(\partial i)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i} + \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i}}{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial i)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i} + \tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i}}. \tag{16}$$

Neglecting higher order cumulants

Montanari and Rizzo proceed by neglecting cavity cumulants $\mathcal{C}_{\mathcal{A}}^{\backslash i}$ with $|\mathcal{A}| > 2$. Denote by $\mathrm{Part}_2(\mathcal{A})$ the set of all partitions of $\mathcal{A}$ into subsets which have cardinality 2 at most. Thus, neglecting higher order cavity cumulants amounts to the following approximation:

$$\mathcal{M}_{\mathcal{A}}^{\backslash i} \approx \sum_{\mathcal{B}\in\mathrm{Part}_2(\mathcal{A})} \prod_{\mathcal{E}\in\mathcal{B}} \mathcal{C}_{\mathcal{E}}^{\backslash i}. \tag{17}$$

By some algebraic manupulations, one can express the consistency equations (14) = (15) in this approximation as follows:

$$\begin{aligned}\mathcal{M}_j^{\backslash i} = &\frac{\tanh\theta_j \sum_{\mathcal{A}\in\mathcal{P}_+(\partial j\backslash i)} t_{j\mathcal{B}} \mathcal{M}_{\mathcal{B}}^{\backslash j} + \sum_{\mathcal{A}\in\mathcal{P}_-(\partial j\backslash i)} t_{j\mathcal{B}} \mathcal{M}_{\mathcal{B}}^{\backslash j}}{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial j\backslash i)} t_{j\mathcal{B}} \mathcal{M}_{\mathcal{B}}^{\backslash j} + \tanh\theta_j \sum_{\mathcal{A}\in\mathcal{P}_-(\partial j\backslash i)} t_{j\mathcal{B}} \mathcal{M}_{\mathcal{B}}^{\backslash j}} \\ &- \sum_{k\in\partial i\backslash j} t_{ik} \mathcal{C}_{jk}^{\backslash i} \frac{\tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_+(\partial i\backslash\{j,k\})} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i} + \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i\backslash\{j,k\})} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i}}{\sum_{\mathcal{A}\in\mathcal{P}_+(\partial i\backslash j)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i} + \tanh\theta_i \sum_{\mathcal{A}\in\mathcal{P}_-(\partial i\backslash j)} t_{i\mathcal{A}} \mathcal{M}_{\mathcal{A}}^{\backslash i}}\end{aligned} \tag{18}$$

One can use (17) to write (18) in terms of the singleton cumulants $\{\mathcal{M}_j^{\backslash i}\}_{i\in\mathcal{V},j\in\partial i}$ and the pair cumulants $\{\mathcal{C}_{jk}^{\backslash i}\}_{i\in\mathcal{V},j\in\partial i,k\in\partial i\backslash j}$. Given (estimates of) the pair cumulants, the consistency equations (18) are thus fixed point equations in the singleton cumulants.

The procedure is now:

- Estimate the pair cumulants $\{\mathcal{C}_{jk}^{\backslash i}\}_{i\in\mathcal{V},j\in\partial i,k\in\partial i\backslash j}$ using BP in combination with linear response (called "response propagation" in Montanari and Rizzo (2005)).
- Calculate the fixed point $\{\mathcal{M}_j^{\backslash i}\}_{i\in\mathcal{V},j\in\partial i}$ of (18) using the estimated the pair cumulants.
- Use (16) in combination with (17) to calculate the final expectation values $\{M_j\}_{j\in\mathcal{V}}$ using the estimated pair cumulants and the fixed point of (18).

Linearized version

The update equations can be linearized by expanding up to first order in the pair cumulants $\mathcal{C}_{jk}^{\backslash i}$. This yields the following linearized consistency equation (Montanari and Rizzo, 2005):

$$M_j^{(i)} = T_i^{\backslash j} - \sum_{l \in \partial i \backslash j} \Omega_{j,l}^{(i)} t_{il} C_{jl}^{(i)} + \sum_{\{l_1,l_2\}: l_1,l_2 \in \partial j \backslash i} \Gamma_{i,l_1 l_2}^{\backslash j} t_{jl_1} t_{jl_2} C_{l_1 l_2}^{(j)} \tag{19}$$

where

$$T_{\mathcal{A}}^{\backslash i} := \tanh\left(\theta_i + \sum_{k \in \partial i \backslash \mathcal{A}} \tanh^{-1}\left(t_{ik} M_k^{(i)}\right)\right)$$

$$\Omega_{jl}^{\backslash i} := \frac{T_{jl}^{\backslash i}}{1 + t_{il} M_l^{\backslash i} T_{jl}^{\backslash i}}$$

$$\Gamma_{i,l_1 l_2}^{\backslash j} := \frac{T_{il_1 l_2}^{\backslash j} - T_i^{\backslash j}}{1 + t_{jl_1} t_{jl_2} M_{l_1}^{(j)} M_{l_2}^{(j)} + t_{jl_1} M_{l_1}^{(j)} T_{il_1 l_2}^{\backslash j} + t_{jl_2} M_{l_2}^{(j)} T_{il_1 l_2}^{\backslash j}}.$$

The final magnetizations (16) are, up to first order in the pair cumulants:

$$M_j = T^{\backslash j} + \sum_{\{l_1,l_2\}: l_1,l_2 \in \partial j^2} \Gamma_{l_1 l_2}^{\backslash j} t_{jl_1} t_{jl_2} C_{l_1 l_2}^{(j)} + \mathcal{O}(C^2)$$

where

$$\Gamma_{l_1 l_2}^{\backslash j} := \frac{T_{l_1 l_2}^{\backslash j} - T^{\backslash j}}{1 + t_{jl_1} t_{jl_2} M_{l_1}^{\backslash j} M_{l_2}^{(j)} + t_{jl_1} M_{l_1}^{(j)} T_{l_1 l_2}^{\backslash j} + t_{jl_2} M_{l_2}^{(j)} T_{l_1 l_2}^{\backslash j}}.$$

References

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

Michael Chertkov and Vladimir Y Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009, 2006. URL `http://stacks.iop.org/1742-5468/2006/P06009`.

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.

V. Gómez, J. Mooij, and H. Kappen. *In preparation*, 2006.

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.

Tommi Jaakkola and Michael I. Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999. URL `http://www.jair.org/papers/paper583.html`.

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

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the Sum-Product Algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

M. Mézard, G. Parisi, and M. A. Virasoro. *Spin glass theory and beyond*. World Scientific, Singapore, 1987.

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.

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.

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. URL `http://stacks.iop.org/1742-5468/2005/P10011`.

J M Mooij and H J Kappen. On the properties of the bethe approximation and loopy belief propagation on binary networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11012, 2005. URL `http://stacks.iop.org/1742-5468/2005/P11012`.

G. Parisi. *Statistical Field Theory*. Addison-Wesley, Redwood City, Ca, 1988.

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

J. Pearl. *Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988.

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

M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann, and G. F. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference algorithms. *Methods of information in Medicine*, 30(4):241–255, October 1991.

Masami Takikawa and Bruce D'Ambrosio. Multiplicative factorization of noisy-max. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 622–63, San Francisco, CA, 1999. Morgan Kaufmann.

Max Welling and Yee Whye Teh. Linear response for approximate inference. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

Max Welling, Thomas Minka, and Yee Whye Teh. Structured Region Graphs: Morphing EP into GBP. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 609, Arlington, Virginia, 2005. AUAI Press.

W. Wiegerinck, H. J. Kappen, E. W. M. T. ter Braak, W. J. P. P. ter Burg, M. J. Nijman, Y. L. O, and J. P. Neijt. Approximate inference for medical diagnosis. *Pattern Recognition Letters*, 20:1231–1239, 1999.

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.

A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.

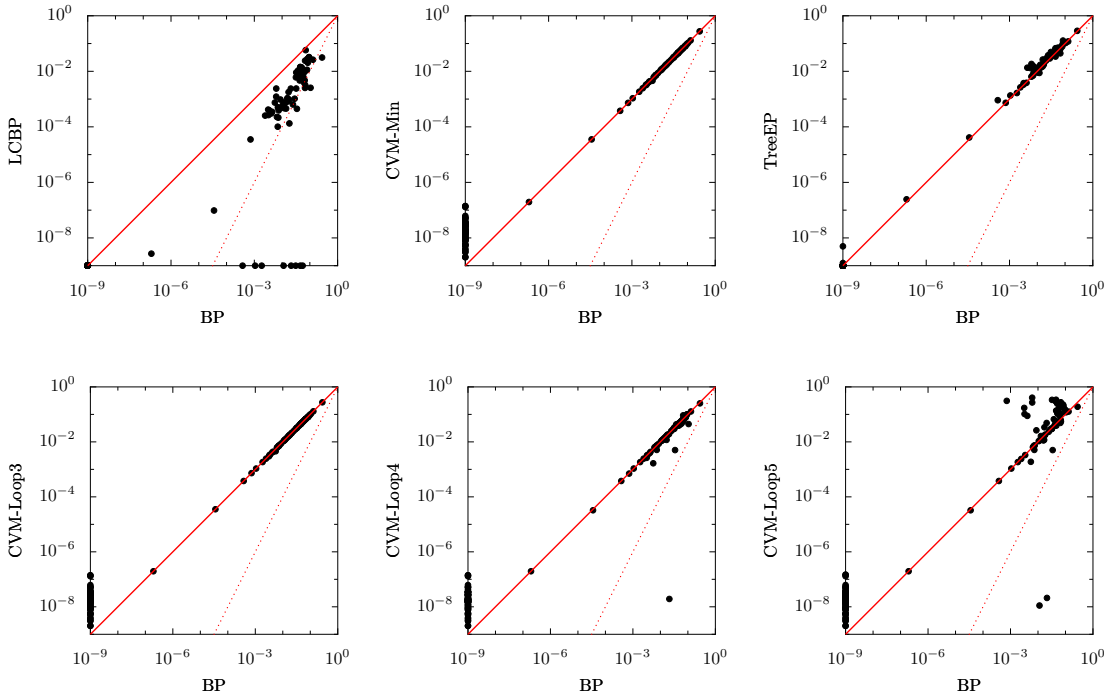


Figure 13: Scatter plots of errors for PROMEDAS instances.

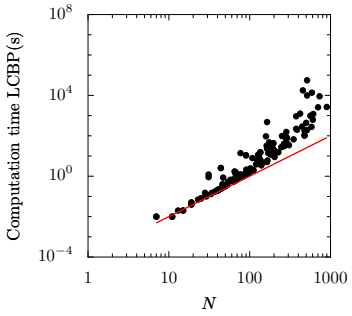


Figure 14: Computation time (in seconds) of LCBP for PROMEDAS instances vs. N, the number of variables in the preprocessed graphical model. The solid line corresponds to $t \propto N^2$.