Novel Bounds on Marginal Probabilities

Joris M. Mooij joris.mooij@tuebingen.mpg.de
MPI for Biological Cybernetics, Dept. Schölkopf
Spemannstraße 38, 72076 Tübingen, Germany

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

Editor:

Abstract

We derive two related novel bounds on single-variable marginal probability distributions in factor graphs with discrete variables. The first method propagates bounds over a subtree of the factor graph rooted in the variable, and the second method propagates bounds over the self-avoiding walk tree starting at the variable. By construction, both methods not only bound the exact marginal probability distribution of a variable, but also its approximate Belief Propagation marginal ("belief"). Thus, apart from providing a practical means to calculate bounds on marginals, our contribution also lies in an increased understanding of the error made by Belief Propagation. Empirically, we show that our bounds often outperform existing bounds in terms of accuracy and/or computation time. We also show that our bounds can yield nontrivial results for medical diagnosis inference problems.

Keywords: Graphical Models, Factor Graphs, Inference, Marginal Probability Distributions, Bounds

1. Introduction

Graphical models are used in many different fields. A fundamental problem in the application of graphical models is that exact inference is NP-hard (Cooper, 1990). In recent years, much research has focused on approximate inference techniques, such as sampling methods and deterministic approximation methods, e.g., Belief Propagation (BP) (Pearl, 1988). Although the approximations obtained by these methods can be very accurate, there are only few guarantees on the error of the approximation, and often it is not known (without comparing with the exact solution) how accurate an approximate result is. Thus it is desirable to calculate, in addition to the approximate results, tight bounds on the approximation error. Existing methods to calculate bounds on marginals include (Tatikonda, 2003; Leisink and Kappen, 2003; Taga and Mase, 2006; Ihler, 2007). Also, upper bounds on the partition sum, e.g., (Jaakkola and Jordan, 1996; Wainwright et al., 2005), can be combined with lower bounds on the partition sum, such as the well-known mean field bound or higher-order lower bounds (Leisink and Kappen, 2001), to obtain bounds on marginals.

In this article, we derive novel bounds on exact single-variable marginals in factor graphs. The original motivation for this work was to better understand and quantify the BP error.

This has lead to bounds which are at the same time bounds for the exact single-variable marginals as well as for the BP beliefs. A particularly nice feature of the bounds is that their computational cost is relatively low, provided that the number of possible values of each variable in the factor graph is small. Unfortunately, the computation time is exponential in the number of possible values of the variables, which limits application to factor graphs in which each variable has a low number of possible values. On these factor graphs however, our bounds perform exceedingly well and we show empirically that they outperform the state-of-the-art in a variety of factor graphs, including real-world problems arising in medical diagnosis.

This article is organized as follows. In the next section, we derive our novel bounds. In Section 3, we discuss related work. In Section 4 we present experimental results. We conclude with conclusions and a discussion in Section 5.

2. Theory

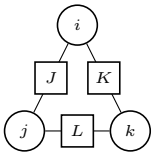
In this work, we consider graphical models such as Markov random fields and Bayesian networks. We use the unifying factor graph representation (Kschischang et al., 2001). In the first subsection, we introduce our notation and some basic definitions concerning factor graphs. Then, we shortly remind the reader of some basic facts about convexity. After that, we introduce some notation and concepts for measures on subsets of variables. We proceed with a subsection that considers the interplay between convexity and the operations of normalization and multiplication. In the next subsection, we introduce "(smallest bounding) boxes" that will be used to describe sets of measures in a convenient way. Then, we formulate the basic lemma that will be used to obtain bounds on marginals. We illustrate the basic lemma with two simple examples. Then we formulate our first result, an algorithm for propagating boxes over a subtree of the factor graph, which results in a bound on the marginal of the root variable of the subtree. In the last subsection, we show how one can go deeper into the computation tree and derive our second result, an algorithm for propagating boxes over self-avoiding walk trees. The result of that algorithm is a bound on the marginal of the root variable (starting point) of the self-avoiding walk tree. For the special case where all factors in the factor graph depend on two variables at most ("pairwise interactions"), our first result is equivalent to a truncation of the second one. This is not true for higher-order interactions, however.

2.1 Factor graphs

Let $\mathcal{V} := \{1, \ldots, N\}$ and consider N discrete random variables $(x_i)_{i\in\mathcal{V}}$. Each variable x_i takes values in a discrete domain $\mathcal{X}_i$. We will frequently use the following multi-index notation. Let $A = \{i_1, i_2, \ldots, i_m\} \subseteq \mathcal{V}$ with $i_1 < i_2 < \ldots i_m$. We write $\mathcal{X}_A := \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_m}$ and for any family $(Y_i)_{i\in B}$ with $A \subseteq B \subseteq \mathcal{V}$, we write $Y_A := (Y_{i_1}, Y_{i_2}, \ldots, Y_{i_m})$.

We consider a probability distribution over $x = (x_1, \ldots, x_N) \in \mathcal{X}_\mathcal{V}$ that can be written as a product of factors (also called "interactions") $(\psi_I)_{I\in\mathcal{F}}$:

$$\mathbb{P}(x) = \frac{1}{Z} \prod_{I\in\mathcal{F}} \psi_I(x_{N_I}), \qquad Z = \sum_{x\in\mathcal{X}_\mathcal{V}} \prod_{I\in\mathcal{F}} \psi_I(x_{N_I}). \tag{1}$$



$$\mathbb{P}(x_i, x_j, x_k) = \tfrac{1}{Z}\psi_J(x_i, x_j)\psi_K(x_i, x_k)\psi_L(x_j, x_k)$$

$$Z = \sum_{x_i \in \mathcal{X}_i} \sum_{x_j \in \mathcal{X}_j} \sum_{x_k \in \mathcal{X}_k} \psi_J(x_i, x_j)\psi_K(x_i, x_k)\psi_L(x_j, x_k)$$

Figure 1: Example of a factor graph with three variable nodes (i, j, k), represented as circles, and three factor nodes (J, K, L), represented as rectangles. The corresponding random variables are x_i, x_j, x_k; the corresponding factors are $\psi_J : \mathcal{X}_i \times \mathcal{X}_j \to [0, \infty)$, $\psi_K : \mathcal{X}_i \times \mathcal{X}_k \to [0, \infty)$ and $\psi_L : \mathcal{X}_j \times \mathcal{X}_k \to [0, \infty)$. The corresponding probability distribution $\mathbb{P}(x)$ is written out on the right.

For each factor index $I \in \mathcal{F}$, there is an associated subset $N_I \subseteq \mathcal{V}$ of variable indices and the factor ψ_I is a nonnegative function $\psi_I : \mathcal{X}_{N_I} \to [0, \infty)$. For a Bayesian network, the factors are (conditional) probability tables. In case of Markov random fields, the factors are often called potentials.[1] In the following, we will use lowercase for variable indices and uppercase for factor indices.

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 $\mathbb{P}(x_A)$ for subsets of variables $A \subseteq \mathcal{V}$ is intractable in general. In this article, we focus on the task of obtaining rigorous bounds on single-variable marginals $\mathbb{P}(x_i) = \sum_{x_{\mathcal{V}\setminus\{i\}}} \mathbb{P}(x)$.

We can represent the structure of the probability distribution (1) using a *factor graph* $(\mathcal{V}, \mathcal{F}, \mathcal{E})$. This is a bipartite graph, consisting of *variable nodes* $i \in \mathcal{V}$, *factor nodes* $I \in \mathcal{F}$, and *edges* $e \in \mathcal{E}$, with an edge $\{i, I\}$ between $i \in \mathcal{V}$ and $I \in \mathcal{F}$ if and only if the factor ψ_I depends on x_i (i.e., if $i \in N_I$). We will represent factor nodes visually as rectangles and variable nodes as circles. Figure 1 shows a simple example of a factor graph and the corresponding probability distribution. The set of neighbors of a factor node I is precisely N_I; similarly, we denote the set of neighbors of a variable node i by $N_i := \{I \in \mathcal{F} : i \in N_I\}$. 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.

We will assume throughout this article that the factor graph corresponding to (1) is connected. Furthermore, we will assume that

$$\forall I \in \mathcal{F}\ \forall i \in N_I\ \forall x_{N_I\setminus\{i\}} \in \mathcal{X}_{N_I\setminus\{i\}} : \sum_{x_i \in \mathcal{X}_i} \psi_I(x_i, x_{N_I\setminus\{i\}}) > 0.$$

This will prevent technical problems regarding normalization later on.[2]

One final remark concerning notation: we will sometimes abbreviate $\{i\}$ as i if no confusion can arise.

1. Not to be confused with statistical physics terminology, where "potential" refers to $-\frac{1}{\beta}\log\psi_I$ instead, with β the inverse temperature.
2. This condition ensures that if one runs Belief Propagation on the factor graph, the messages will always remain nonzero, provided that the initial messages are nonzero.

2.2 Convexity

Let V be a real vector space. For T elements $(v_t)_{t=1,\ldots,T}$ of V and T nonnegative numbers $(\lambda_t)_{t=1,\ldots,T}$ with $\sum_{t=1}^T \lambda_t = 1$, we call $\sum_{t=1}^T \lambda_t v_t$ a *convex combination* of the $(v_t)_{t=1,\ldots,T}$ with weights $(\lambda_t)_{t=1,\ldots,T}$. A subset $X \subseteq V$ is called *convex* if for all $x_1, x_2 \in X$ and all $\lambda \in [0,1]$, the convex combination $\lambda x_1 + (1-\lambda)x_2 \in X$. An *extreme point* of a convex set X is an element $x \in X$ which cannot be written as a (nontrivial) convex combination of two different points in X. In other words, $x \in X$ is an extreme point of X if and only if for all $\lambda \in (0,1)$ and all $x_1, x_2 \in X$, $x = \lambda x_1 + (1-\lambda)x_2$ implies $x_1 = x_2$. We denote the set of extreme points of a convex set X by $\mathrm{Ext}\,(X)$. For a subset Y of the vector space V, we define the *convex hull of* Y to be the smallest convex set $X \subseteq V$ with $Y \subseteq X$; we denote the convex hull of Y as $\mathrm{Hull}\,(Y)$.

2.3 Measures and operators

For $A \subseteq \mathcal{V}$, define $\mathcal{M}_A := [0,\infty)^{\mathcal{X}_A}$, i.e., $\mathcal{M}_A$ is the set of nonnegative functions on $\mathcal{X}_A$. $\mathcal{M}_A$ can be identified with the set of finite measures on $\mathcal{X}_A$. We will simply call the elements of $\mathcal{M}_A$ "measures on A". We also define $\mathcal{M}_A^* := \mathcal{M}_A \setminus \{0\}$. We will denote $\mathcal{M} := \bigcup_{A\subseteq\mathcal{V}} \mathcal{M}_A$ and $\mathcal{M}^* := \bigcup_{A\subseteq\mathcal{V}} \mathcal{M}_A^*$.

Adding two measures $\Psi, \Phi \in \mathcal{M}_A$ results in the measure $\Psi + \Phi$ in $\mathcal{M}_A$. For $A, B \subseteq \mathcal{V}$, we can multiply an element of $\mathcal{M}_A$ with an element of $\mathcal{M}_B$ to obtain an element of $\mathcal{M}_{A\cup B}$; a special case is multiplication with a scalar. Note that there is a natural embedding of $\mathcal{M}_A$ in $\mathcal{M}_B$ for $A \subseteq B \subseteq \mathcal{V}$ obtained by multiplying an element $\Psi \in \mathcal{M}_A$ by $\mathbf{1}_{B\setminus A} \in \mathcal{M}_{B\setminus A}$, the constant function with value 1 on $\mathcal{X}_{B\setminus A}$. Another important operation is the partial summation: given $A \subseteq B \subseteq \mathcal{V}$ and $\Psi \in \mathcal{M}_B$, define $\sum_{x_A} \Psi$ to be the measure in $\mathcal{M}_{B\setminus A}$ that satisfies

$$\left(\sum_{x_A} \Psi\right)(x_{B\setminus A}) = \sum_{x_A \in \mathcal{X}_A} \Psi(x_A, x_{B\setminus A}) \qquad \forall x_{B\setminus A} \in \mathcal{X}_{B\setminus A}.$$

Also, defining $A' = B \setminus A$, we will sometimes write this measure as $\sum_{x_{\setminus A'}} \Psi$, which is an abbreviation of $\sum_{x_{B\setminus A'}} \Psi$. This notation does not make explicit which variables are summed over (which depends on the measure that is being partially summed), although it shows which variables remain after summation.

In the following, we will implicitly define operations on *sets* of measures by applying the operation on elements of these sets and taking the set of the resulting measures; e.g., if we have two subsets $\Xi_A \subseteq \mathcal{M}_A$ and $\Xi_B \subseteq \mathcal{M}_B$ for $A, B \subseteq \mathcal{V}$, we define the product of the sets Ξ_A and Ξ_B to be the set of the products of elements of Ξ_A and Ξ_B, i.e., $\Xi_A\Xi_B := \{\Psi_A\Psi_B : \Psi_A \in \Xi_A, \Psi_B \in \Xi_B\}$.

In Figure 2, the simple case of a binary random variable x_i and the subset $A = \{i\}$ is illustrated. Note that in this case, a measure $\Psi \in \mathcal{M}_i$ can be identified with a point in the quarter plane $[0,\infty) \times [0,\infty)$.

We will define $\mathcal{Q}_A$ to be the set of completely factorized measures on A, i.e.,

$$\mathcal{Q}_A := \prod_{a\in A} \mathcal{M}_{\{a\}} = \left\{ \prod_{a\in A} \Psi_a : \Psi_a \in \mathcal{M}_{\{a\}} \text{ for each } a \in A \right\}.$$

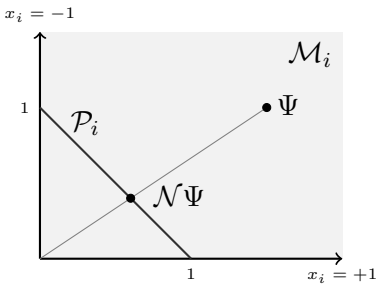


Figure 2: Illustration of some concepts in the simple case of a binary random variable $x_i \in \mathcal{X}_i = \{\pm 1\}$ and the subset $A = \{i\}$. A measure $\Psi \in \mathcal{M}_i$ can be identified with a point in the quarter plane as indicated in the Figure. A normalized measure can be obtained by scaling Ψ; the result $\mathcal{N}\Psi$ is contained in the simplex $\mathcal{P}_i$, a lower-dimensional submanifold of $\mathcal{M}_i$.

Note that $\mathcal{M}_A$ is the convex hull of $\mathcal{Q}_A$. Indeed, we can write each measure $\Psi \in \mathcal{M}_A$ as a convex combination of measures in $\mathcal{Q}_A$; let $Z := \sum_{x_A} \Psi$ and note that

$$\Psi(x) = \sum_{y \in \mathcal{X}_A} \frac{\Psi(y)}{Z} \left(Z \delta_y(x)\right) \qquad \forall x \in \mathcal{X}_A.$$

For any $y \in \mathcal{X}_A$, the Kronecker delta function $\delta_y \in \mathcal{M}_A$ (which is 1 if its argument is equal to y and 0 otherwise) is an element of $\mathcal{Q}_A$ because $\delta_y(x) = \prod_{a \in A} \delta_{y_a}(x_a)$. We denote $\mathcal{Q}_A^* := \mathcal{Q}_A \setminus \{0\}$.

We define the *partition sum operator* $\mathcal{Z} : \mathcal{M} \to [0, \infty)$ which calculates the partition sum (normalization constant) of a measure, i.e.,

$$\mathcal{Z}\Psi := \sum_{x_A \in \mathcal{X}_A} \Psi(x_A) \qquad \text{for } \Psi \in \mathcal{M}_A,\ A \subseteq \mathcal{V}.$$

We denote with $\mathcal{P}_A$ the set of probability measures on A, i.e., $\mathcal{P}_A = \{\Psi \in \mathcal{M}_A : \mathcal{Z}\Psi = 1\}$, and define $\mathcal{P} := \bigcup_{A \subseteq \mathcal{V}} \mathcal{P}_A$. The set $\mathcal{P}_A$ is called a *simplex* (see also Figure 2). Note that a simplex is convex; the simplex $\mathcal{P}_A$ has precisely $\#(\mathcal{X}_A)$ extreme points, each of which corresponds to putting all probability mass on one of the possible values of x_A.

Define the *normalization operator* $\mathcal{N} : \mathcal{M}^* \to \mathcal{P}$ which normalizes a measure, i.e.,

$$\mathcal{N}\Psi := \frac{1}{\mathcal{Z}\Psi}\Psi \qquad \text{for } \Psi \in \mathcal{M}^*.$$

Note that $\mathcal{Z} \circ \mathcal{N} = 1$. Figure 2 illustrates the normalization of a measure in a simple case.

2.4 Convex sets of measures

To calculate marginals of subsets of variables in some factor graph, several operations performed on measures are relevant: normalization, taking products of measures, and summing over subsets of variables. In this section we study the interplay between convexity and these

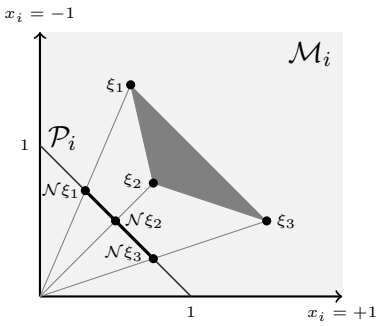


Figure 3: Any convex combination of $\mathcal{N}\xi_1$, $\mathcal{N}\xi_2$ and $\mathcal{N}\xi_3$ can be written as a normalized convex combination of ξ_1, ξ_2 and ξ_3. Vice versa, normalizing a convex combination of ξ_1, ξ_2 and ξ_3 yields a convex combination of $\mathcal{N}\xi_1, \mathcal{N}\xi_2$ and $\mathcal{N}\xi_3$.

operations; this will turn out to be useful later on, because our bounds make use of convex sets of measures that are propagated over the factor graph.

The interplay between normalization and convexity is described by the following Lemma, which is illustrated in Figure 3.

Lemma 1 *Let $A \subseteq \mathcal{V}$, $T \in \mathbb{N}^*$ and let $(\xi_t)_{t=1,\ldots,T}$ be elements of $\mathcal{M}_A^*$. Each convex combination of the normalized measures $(\mathcal{N}\xi_t)_{t=1,\ldots,T}$ can be written as a normalized convex combination of the measures $(\xi_t)_{t=1,\ldots,T}$ (which has different weights in general), and vice versa.*

Proof Let $(\lambda_t)_{t=1,\ldots,T}$ be nonnegative numbers with $\sum_{t=1}^T \lambda_t = 1$. Then

$$\mathcal{Z}\left(\sum_{t=1}^T \lambda_t \mathcal{N}\xi_t\right) = \sum_{t=1}^T \lambda_t \mathcal{Z}\mathcal{N}\xi_t = 1,$$

therefore

$$\sum_{t=1}^T \lambda_t(\mathcal{N}\xi_t) = \mathcal{N}\left(\sum_{t=1}^T \lambda_t(\mathcal{N}\xi_t)\right) = \mathcal{N}\left(\sum_{t=1}^T \frac{\lambda_t}{\mathcal{Z}\xi_t}\xi_t\right) = \mathcal{N}\left(\sum_{t=1}^T \frac{\frac{\lambda_t}{\mathcal{Z}\xi_t}}{\sum_{s=1}^T \frac{\lambda_s}{\mathcal{Z}\xi_s}}\,\xi_t\right),$$

which is the result of applying the normalization operator to a convex combination of the elements $(\xi_t)_{t=1,\ldots,T}$.

Vice versa, let $(\mu_t)_{t=1,\ldots,T}$ be nonnegative numbers with $\sum_{t=1}^T \mu_t = 1$. Then

$$\mathcal{N}\left(\sum_{t=1}^T \mu_t \xi_t\right) = \sum_{t=1}^T \frac{\mu_t}{Z}\xi_t$$

where

$$Z := \mathcal{Z}\left(\sum_{t=1}^T \mu_t \xi_t\right) = \sum_{t=1}^T \mu_t \mathcal{Z}\xi_t = \sum_{t=1}^T \mu_t Z_t$$

where we defined $Z_t := \mathcal{Z}\xi_t$ for all $t = 1, \ldots, T$. Thus

$$\mathcal{N}\left(\sum_{t=1}^{T} \mu_t \xi_t\right) = \sum_{t=1}^{T} \frac{\mu_t}{\sum_{s=1}^{T} \mu_s Z_s} \xi_t = \sum_{t=1}^{T} \frac{\mu_t Z_t}{\sum_{s=1}^{T} \mu_s Z_s} \mathcal{N}\xi_t,$$

which is a convex combination of the normalized measures $(\mathcal{N}\xi_t)_{t=1,\ldots,T}$. ■

The following lemma concerns the interplay between convexity and taking products; it says that if we take the product of convex sets of measures on different spaces, the resulting set is contained in the convex hull of the product of the extreme points of the convex sets. We have not made a picture corresponding to this lemma because the simplest nontrivial case would require at least four dimensions.

Lemma 2 *Let $T \in \mathbb{N}^*$ and $(A_t)_{t=1,\ldots,T}$ be a family of mutually disjoint subsets of $\mathcal{V}$. For each $t = 1, \ldots, T$, let $\Xi_t \subseteq \mathcal{M}_{A_t}$ be convex with a finite number of extreme points. Then:*

$$\prod_{t=1}^{T} \Xi_t \subseteq \text{Hull}\left(\prod_{t=1}^{T} \text{Ext}\, \Xi_t\right),$$

Proof Let $\Psi_t \in \Xi_t$ for each $t = 1, \ldots, T$. For each t, Ψ_t can be written as a convex combination

$$\Psi_t = \sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} \xi_t, \qquad \sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} = 1, \qquad \forall\ \xi_t \in \text{Ext}\,(\Xi_t) : \lambda_{t;\xi_t} \geq 0.$$

Therefore the product $\prod_{t=1}^{T} \Psi_t$ is also a convex combination:

$$\begin{aligned}
\prod_{t=1}^{T} \Psi_t &= \prod_{t=1}^{T} \left(\sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} \xi_t \right) \\
&= \sum_{\xi_1 \in \text{Ext}\,(\Xi_1)} \sum_{\xi_2 \in \text{Ext}\,(\Xi_2)} \cdots \sum_{\xi_T \in \text{Ext}\,(\Xi_T)} \left(\prod_{t=1}^{T} \lambda_{t;\xi_t} \right) \left(\prod_{t=1}^{T} \xi_t \right) \\
&\in \text{Hull}\left(\prod_{t=1}^{T} \text{Ext}\, \Xi_t \right).
\end{aligned}$$

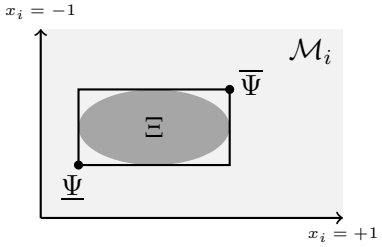
■

2.5 Boxes and smallest bounding boxes

In this subsection, we define "(smallest bounding) boxes", certain convex sets of measures that will play a central role in our bounds, and study some of their properties.

Definition 3 *Let $A \subseteq \mathcal{V}$. For $\underline{\Psi} \in \mathcal{M}_A$ and $\overline{\Psi} \in \mathcal{M}_A$ with $\underline{\Psi} \leq \overline{\Psi}$, we define the* box *between the lower bound $\underline{\Psi}$ and the upper bound $\overline{\Psi}$ by*

$$\mathcal{B}_A\left(\underline{\Psi}, \overline{\Psi}\right) := \{\Psi \in \mathcal{M}_A : \underline{\Psi} \leq \Psi \leq \overline{\Psi}\}.$$



$$\mathcal{B}(\Xi) = \mathcal{B}_i\left(\underline{\Psi}, \overline{\Psi}\right)$$

Figure 4: The smallest bounding box $\mathcal{B}(\Xi)$ for Ξ is given by the box $\mathcal{B}_i\left(\underline{\Psi}, \overline{\Psi}\right)$ with lower bound $\underline{\Psi}$ and upper bound $\overline{\Psi}$.

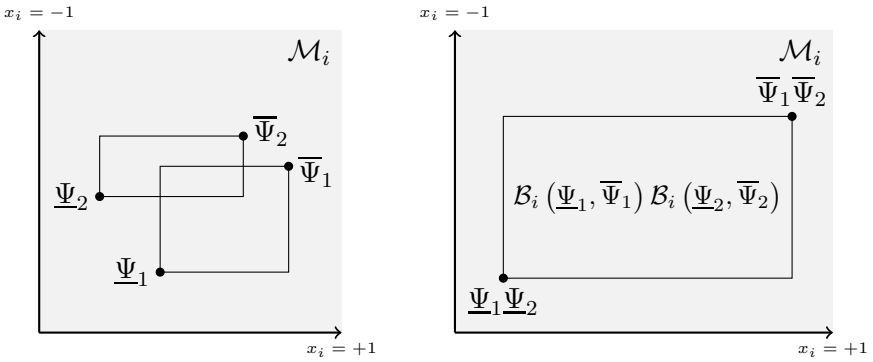


Figure 5: Multiplication of two boxes on the same variable set $A = \{i\}$.

The inequalities should be interpreted pointwise, e.g., $\underline{\Psi} \leq \Psi$ means $\underline{\Psi}(x) \leq \Psi(x)$ for all $x \in \mathcal{X}_A$. Note that a box is convex; indeed, its extreme points are the "corners" of which there are $2^{\#(\mathcal{X}_A)}$.

Definition 4 *Let $A \subseteq \mathcal{V}$ and $\Xi \subseteq \mathcal{M}_A$ be bounded (i.e., $\Xi \leq \Psi$ for some $\Psi \in \mathcal{M}_A$). The* smallest bounding box *for Ξ is defined as $\mathcal{B}(\Xi) := \mathcal{B}_A\left(\underline{\Psi}, \overline{\Psi}\right)$, where $\underline{\Psi}, \overline{\Psi} \in \mathcal{M}_A$ are given by*

$$\begin{aligned} \underline{\Psi}(x_A) &:= \inf\{\Psi(x_A) : \Psi \in \Xi\} \qquad \forall x_a \in \mathcal{X}_A, \\ \overline{\Psi}(x_A) &:= \sup\{\Psi(x_A) : \Psi \in \Xi\} \qquad \forall x_a \in \mathcal{X}_A. \end{aligned}$$

Figure 4 illustrates this concept. Note that $\mathcal{B}(\Xi) = \mathcal{B}(\text{Hull}(\Xi))$. Therefore, if Ξ is convex, the smallest bounding box for Ξ depends only on the extreme points $\text{Ext}(\Xi)$, i.e., $\mathcal{B}(\Xi) = \mathcal{B}(\text{Ext}(\Xi))$.

The product of several boxes on the same subset A of variables can be easily calculated as follows (see also Figure 5).

Lemma 5 *Let $A \subseteq \mathcal{V}$, $T \in \mathbb{N}^*$ and for each $t = 1, \ldots, T$, let $\underline{\Psi}_t, \overline{\Psi}_t \in \mathcal{M}_A$ such that $\underline{\Psi}_t \leq \overline{\Psi}_t$. Then*

$$\prod_{t=1}^{T} \mathcal{B}_A\left(\underline{\Psi}_t, \overline{\Psi}_t\right) = \mathcal{B}_A\left(\prod_{t=1}^{T} \underline{\Psi}_t, \prod_{t=1}^{T} \overline{\Psi}_t\right),$$

i.e., the product of the boxes is again a box, with as lower bound the product of the lower bounds of the boxes and as upper bound the product of the upper bounds of the boxes.

Proof We prove the case $T = 2$; the general case follows by induction. We show that

$$\mathcal{B}_A\left(\underline{\Psi}_1, \overline{\Psi}_1\right) \mathcal{B}_A\left(\underline{\Psi}_2, \overline{\Psi}_2\right) = \mathcal{B}_A\left(\underline{\Psi}_1\,\underline{\Psi}_2, \overline{\Psi}_1\,\overline{\Psi}_2\right).$$

That is, for $\Phi \in \mathcal{M}_A$ we have to show that

$$\underline{\Psi}_1(x)\underline{\Psi}_2(x) \leq \Phi(x) \leq \overline{\Psi}_1(x)\overline{\Psi}_2(x) \qquad \forall x \in \mathcal{X}_A$$

if and only if there exist $\Phi_1, \Phi_2 \in \mathcal{M}_A$ such that:

$$\begin{aligned}
&\Phi(x) = \Phi_1(x)\Phi_2(x) && \forall x \in \mathcal{X}_A;\\
&\underline{\Psi}_1(x) \leq \Phi_1(x) \leq \overline{\Psi}_1(x) && \forall x \in \mathcal{X}_A;\\
&\underline{\Psi}_2(x) \leq \Phi_2(x) \leq \overline{\Psi}_2(x) && \forall x \in \mathcal{X}_A.
\end{aligned}$$

Note that the problem "decouples" for the various possible values of $x \in \mathcal{X}_A$ so that we can treat each component (indexed by $x \in \mathcal{X}_A$) seperately. That is, the problem reduces to showing that

$$[a, b] \cdot [c, d] = [ac, bd]$$

for $0 \leq a \leq b$ and $0 \leq c \leq d$ (take $a = \underline{\Psi}_1(x)$, $b = \overline{\Psi}_1(x)$, $c = \underline{\Psi}_2(x)$ and $d = \overline{\Psi}_2(x)$). In other words, we have to show that $y \in [ac, bd]$ if and only if there exist $y_1 \in [a, b]$, $y_2 \in [c, d]$ with $y = y_1 y_2$. For the less trivial part of this assertion, it is easily verified that choosing y_1 and y_2 according to the following table:

Condition	y_1	y_2
$bc \leq y, b > 0$	b	$\frac{y}{b}$
$b = 0$	0	c
$bc \geq y, c > 0$	$\frac{y}{c}$	c
$bc \geq y, c = 0$	b	0

does the job. ■

In general, the product of several boxes is not a box itself. Indeed, let $i, j \in \mathcal{V}$ be two different variable indices. Then $\mathcal{B}_i\left(\underline{\Psi}_i, \overline{\Psi}_i\right) \mathcal{B}_j\left(\underline{\Psi}_j, \overline{\Psi}_j\right)$ contains only factorizing measures, whereas $\mathcal{B}_{\{i,j\}}\left(\underline{\Psi}_i\underline{\Psi}_j, \overline{\Psi}_i\overline{\Psi}_j\right)$ is not a subset of $\mathcal{Q}_{\{i,j\}}$ in general. However, we do have the following identity:

Lemma 6 *Let $T \in \mathbb{N}^*$ and for each $t = 1, \ldots, T$, let $A_t \subseteq \mathcal{V}$ and $\underline{\Psi}_t, \overline{\Psi}_t \in \mathcal{M}_{A_t}$ such that $\underline{\Psi}_t \leq \overline{\Psi}_t$. Then*

$$\mathcal{B}\left(\prod_{t=1}^{T} \mathcal{B}_{A_t}\left(\underline{\Psi}_t, \overline{\Psi}_t\right)\right) = \mathcal{B}_{\left(\bigcup_{t=1}^{T} A_t\right)}\left(\prod_{t=1}^{T} \underline{\Psi}_t, \prod_{t=1}^{T} \overline{\Psi}_t\right).$$

Proof Straightforward, using the definitions. ■

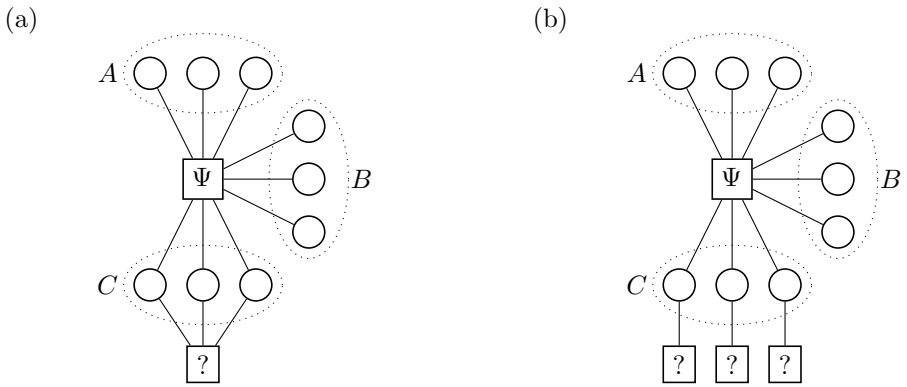


Figure 6: The basic lemma: the smallest bounding box enclosing the set of possible marginals of x_A is identical in cases (a) and (b), if we are allowed to put arbitrary factors on the factor nodes marked with question marks.

2.6 The basic lemma

After defining the elementary concepts, we can proceed with the basic lemma. Given the definitions introduced before, the basic lemma is easy to formulate. It is illustrated in Figure 6.

Lemma 7 *Let $A, B, C \subseteq \mathcal{V}$ be mutually disjoint subsets of variables. Let $\Psi \in \mathcal{M}_{A\cup B\cup C}$ such that for each $x_C \in \mathcal{X}_C$,*

$$\sum_{x_{A\cup B}} \Psi > 0.$$

Then:

$$\mathcal{B}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{M}_C^*\right)\right) = \mathcal{B}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{Q}_C^*\right)\right).$$

Proof Note that $\mathcal{M}_C^*$ is the convex hull of $\mathcal{Q}_C^*$. Furthermore, the multiplication with Ψ and the summation over x_B, x_C preserves convex combinations, as does the normalization operation (see Lemma 1). Therefore,

$$\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{M}_C^*\right) \subseteq \text{Hull}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{Q}_C^*\right)\right)$$

from which the lemma follows. ■

The positivity condition is a technical condition, which in our experience is fulfilled for most practically relevant factor graphs.

2.7 Examples

Before proceeding to the first main result, we first illustrate for a simple case how the basic lemma can be employed to obtain bounds on marginals. We show two bounds for the marginal of the variable x_i in the factor graph in Figure 7(a).

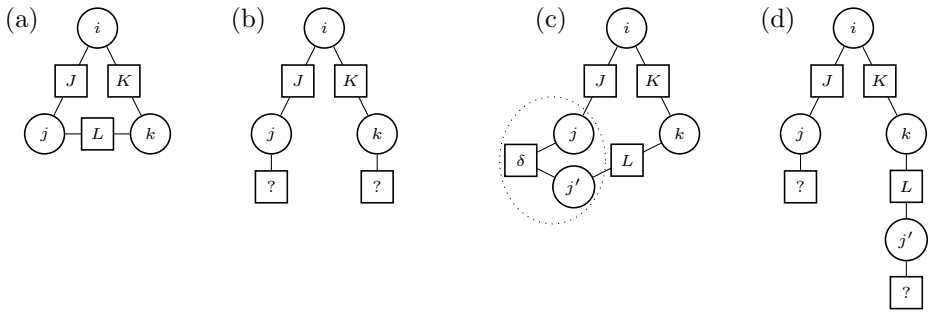


Figure 7: (a) Factor graph; (b) Illustration of the bound on $\mathbb{P}(x_i)$ corresponding to Example I; (c) Cloning node j by adding a new variable j' and a factor $\psi_\delta(x_j, x_{j'}) = \delta_{x_j}(x_{j'})$; (d) Illustration of the improved bound on $\mathbb{P}(x_i)$, corresponding to Example (II), based on (c).

2.7.1 Example I

First, note that the marginal of x_i satisfies

$$\mathbb{P}(x_i) = \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\psi_L\right) \in \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\mathcal{M}^*_{\{j,k\}}\right).$$

because, obviously, $\psi_L \in \mathcal{M}^*_{\{j,k\}}$. Now, applying the basic lemma with $A = \{i\}$, $B = \emptyset$, $C = \{j,k\}$ and $\Psi = \psi_J\psi_K$, we obtain

$$\mathbb{P}(x_i) \in \mathcal{B}\left(\mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\mathcal{Q}^*_{\{j,k\}}\right)\right).$$

Applying the distributive law, we conclude

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\Big(\sum_{x_j}\psi_J\mathcal{M}^*_j\Big)\Big(\sum_{x_k}\psi_K\mathcal{M}^*_k\Big)\right),$$

which certainly implies

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{M}^*_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{M}^*_k\right)\right).$$

This is illustrated in Figure 7(b), which should be read as "What can we say about the range of $\mathbb{P}(x_i)$ when the factors corresponding to the nodes marked with question marks are arbitrary?" Because of the various occurrences of the normalization operator, we can restrict ourselves to normalized measures on the question-marked factor nodes:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{P}_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{P}_k\right)\right).$$

Now it may seem that this smallest bounding box would be difficult to compute, because in principle one would have to compute all the measures in the sets $\mathcal{N}\sum_{x_j}\psi_J\mathcal{P}_j$ and $\mathcal{N}\sum_{x_k}\psi_K\mathcal{P}_k$. Fortunately, we only need to compute the extreme points of these sets, because the mapping

$$\mathcal{M}^*_{\{j\}} \to \mathcal{M}^*_{\{i\}} : \psi \mapsto \mathcal{N}\sum_{x_j}\psi_J\psi$$

maps convex combinations into convex combinations (and similarly for the other mapping, involving ψ_K). Since smallest bounding boxes only depend on extreme points, we conclude that

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J \mathrm{Ext}\,\mathcal{P}_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K \mathrm{Ext}\,\mathcal{P}_k\right)\right)$$

which can be calculated efficiently if the number of possible values of each variable is small.

2.7.2 Example II

We can improve this bound by using another trick: cloning variables. The idea is to first clone the variable x_j by adding a new variable $x_{j'}$ that is constrained to take the same value as x_j. In terms of the factor graph, we add a variable node j' and a factor node δ, connected to variable nodes j and j', with corresponding factor $\psi_\delta(x_j, x_{j'}) := \delta_{x_j}(x_{j'})$; see also Figure 7(c). Clearly, the marginal of x_i satisfies:

$$\begin{aligned}\mathbb{P}(x_i) &= \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\psi_L\right)\\ &= \mathcal{N}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\delta_{x_j}(x_{j'})\right)\end{aligned}$$

where it should be noted that in the first line, ψ_L is shorthand for $\psi_L(x_j, x_k)$ but in the second line it is meant as shorthand for $\psi_L(x_{j'}, x_k)$. Noting that $\psi_\delta \in \mathcal{M}^*_{\{j,j'\}}$ and applying the basic lemma with $C = \{j, j'\}$ yields:

$$\mathbb{P}(x_i) \in \mathcal{N}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\mathcal{M}^*_{\{j,j'\}}\right) \in \mathcal{BN}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\mathcal{Q}^*_{\{j,j'\}}\right).$$

Applying the distributive law, we obtain (see also Figure 7(d)):

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\left(\sum_{x_j}\psi_J\mathcal{M}^*_{\{j\}}\right)\left(\sum_{x_k}\psi_K\sum_{x_{j'}}\psi_L\mathcal{M}^*_{\{j'\}}\right)\right),$$

from which we conclude

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{P}_{\{j\}}\right)\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{BN}\left(\sum_{x_{j'}}\psi_L\mathcal{P}_{\{j'\}}\right)\right)\right).$$

This can again be calculated efficiently by considering only extreme points.

As a more concrete example, take all variables as binary and take for each factor $\psi = \left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right)$. Then the first bound (Example I) yields:

$$\mathbb{P}(x_i) \in \mathcal{B}_i \left(\begin{pmatrix} 1/5 \\ 1/5 \end{pmatrix}, \begin{pmatrix} 4/5 \\ 4/5 \end{pmatrix} \right),$$

whereas the second, tighter, bound (Example II) gives:

$$\mathbb{P}(x_i) \in \mathcal{B}_i \left(\begin{pmatrix} 2/7 \\ 2/7 \end{pmatrix}, \begin{pmatrix} 5/7 \\ 5/7 \end{pmatrix} \right).$$

Obviously, the exact marginal is

$$\mathbb{P}(x_i) = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}.$$

2.8 Propagation of boxes over a subtree

We now formulate a message passing algorithm that resembles Belief Propagation. However, instead of propagating measures, it propagates boxes (or simplices) of measures; furthermore, it is only applied to a subtree of the factor graph, propagating boxes from the leaves towards a root node, instead of propagating iteratively over the whole factor graph several times. The resulting "belief" at the root node is a box that bounds the exact marginal of the root node. The choice of the subtree is arbitrary; different choices lead to different bounds in general. We illustrate the algorithm using the example that we have studied before (see Figure 8).

Definition 8 *Let $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. We call the bipartite graph (V, F, E) a* subtree *of $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ with root i if $i \in V \subseteq \mathcal{V}$, $F \subseteq \mathcal{F}$, $E \subseteq \mathcal{E}$ such that (V, F, E) is a tree with root i and for all $\{j, J\} \in E$, $j \in V$ and $J \in F$ (i.e., there are no "loose edges").*[3]

An illustration of a factor graph and a possible subtree is given in Figure 8(a)-(b). We denote the parent of $j \in V$ according to (V, F, E) by $\mathrm{par}(j)$ and similarly, we denote the parent of $J \in F$ by $\mathrm{par}(J)$. In the following, we will use the topology of the *original* factor graph $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ whenever we refer to neighbors of variables or factors.

Each edge of the subtree will carry one message, oriented such that it "flows" towards the root node. In addition, we define messages entering the subtree for all "missing" edges in the subtree. Because of the bipartite character of the factor graph, we can distinguish between two types of messages: messages $\mathcal{B}_{J \to j} \subseteq \mathcal{M}_j$ sent to a variable $j \in V$ from a neighboring factor $J \in N_j$, and messages $\mathcal{B}_{j \to J} \subseteq \mathcal{M}_j$ sent to a factor $J \in F$ from a neighboring variable $j \in N_J$.

The messages entering the subtree are all defined to be simplices; more precisely, we define the incoming messages

$$\begin{aligned} \mathcal{B}_{j \to J} &= \mathcal{P}_j \qquad J \in F,\, \{j, J\} \in \mathcal{E} \setminus E \\ \mathcal{B}_{J \to j} &= \mathcal{P}_j \qquad j \in V,\, \{j, J\} \in \mathcal{E} \setminus E. \end{aligned}$$

3. Note that this corresponds to the notion of subtree of a bipartite graph; for a subtree of a factor graph, one sometimes imposes the additional constraint that for all factors $J \in F$, all its connecting edges $\{J, j\}$ with $j \in N_J$ have to be in E; here we do not impose this additional constraint.

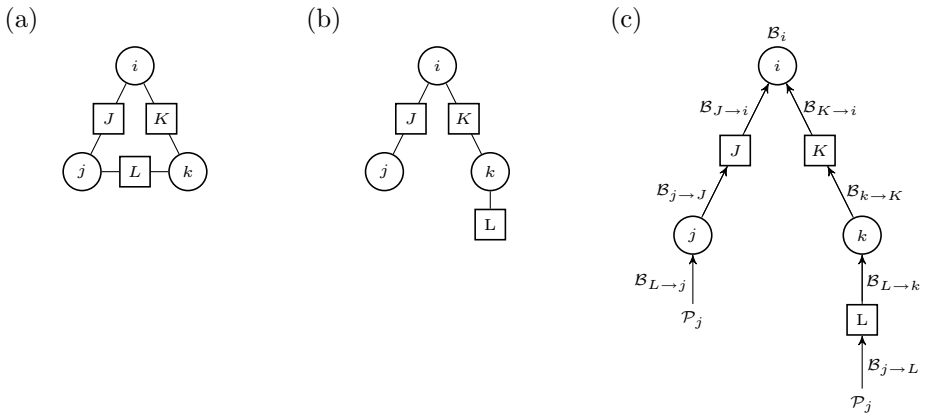


Figure 8: Box propagation algorithm corresponding to Example II: (a) Factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$; (b) a possible subtree (V, F, E) of G; (c) propagating sets of measures (boxes or simplices) on the subtree leading to a bound $\mathcal{B}_i$ on the marginal probability of x_i in $\mathcal{G}$.

We propagate messages towards the root i of the tree using the following update rules (note the similarity with the BP update rules). The message sent from a variable $j \in V$ to its parent $J = \mathrm{par}(j) \in F$ is defined as

$$\mathcal{B}_{j\to J} = \begin{cases} \prod_{K\in N_j\setminus J} \mathcal{B}_{K\to j} & \text{if all incoming } \mathcal{B}_{K\to j} \text{ are boxes} \\ \mathcal{P}_j & \text{if at least one of the } \mathcal{B}_{K\to j} \text{ is the simplex } \mathcal{P}_j, \end{cases}$$

where the product of the boxes can be calculated using Lemma 5. The message sent from a factor $J \in F$ to its parent $k = \mathrm{par}(J) \in V$ is defined as

$$\mathcal{B}_{J\to k} = \mathcal{BN}\left(\sum_{x_{N_J\setminus k}} \psi_J \prod_{l\in N_J\setminus k} \mathcal{B}_{l\to J} \right). \tag{2}$$

This smallest bounding box can be calculated using the following Corollary of Lemma 2:

Corollary 9

$$\mathcal{BN}\left(\sum_{x_{N_J\setminus k}} \psi_J \prod_{l\in N_J\setminus k} \mathcal{B}_{l\to J} \right) = \mathcal{BN}\left(\sum_{x_{N_J\setminus k}} \psi_J \prod_{l\in N_J\setminus k} \mathrm{Ext}\, \mathcal{B}_{l\to J} \right)$$

Proof By Lemma 2,

$$\prod_{l\in N_J\setminus k} \mathcal{B}_{l\to J} \subseteq \mathrm{Hull}\left(\prod_{l\in N_J\setminus k} \mathrm{Ext}\, \mathcal{B}_{l\to J} \right).$$

Because the multiplication with ψ_J and the summation over $x_{N_J \setminus k}$ preserves convex combinations, as does the normalization (see Lemma 1), the statement follows. ■

The final "belief" $\mathcal{B}_i$ at the root node i is calculated by

$$\mathcal{B}_i = \begin{cases} \mathcal{BN}\left(\prod_{K \in N_j} \mathcal{B}_{K \to j}\right) & \text{if all incoming } \mathcal{B}_{K \to j} \text{ are boxes} \\ \mathcal{P}_j & \text{if at least one of the } \mathcal{B}_{K \to j} \text{ is the simplex } \mathcal{P}_j. \end{cases}$$

Note that when applying this to the case illustrated in Figure 8, we obtain the bound that we derived earlier on ("Example II").

We can now formulate our first main result, which gives a rigorous bound on the exact single-variable marginal of the root node:

Theorem 10 *Let $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph with corresponding probability distribution (1). Let $i \in \mathcal{V}$ and (V, F, E) be a subtree of $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ with root $i \in V$. Apply the "box propagation" algorithm described above to calculate the final "belief" $\mathcal{B}_i$ on the root node i. Then $\mathbb{P}(x_i) \in \mathcal{B}_i$.*

Proof sketch The first step consists in extending the subtree such that each factor node has the right number of neighboring variables by cloning the missing variables. The second step consists of applying the basic lemma where the set C consists of all the variable nodes of the subtree which have connecting edges in $\mathcal{E} \setminus E$, together with all the cloned variable nodes. Then we apply the distributive law, which can be done because the extended subtree has no cycles. Finally, we relax the bound by adding additional normalizations and smallest bounding boxes at each factor node in the subtree. It should now be clear that the recursive algorithm "box propagation" described above precisely calculates the smallest bounding box at the root node i that corresponds to this procedure. ■

Note that a subtree of the orginal factor graph is also a subtree of the *computation tree for i* (Tatikonda and Jordan, 2002). A computation tree is an "unwrapping" of the factor graph that has been used in analyses of the Belief Propagation algorithm. The computation tree starting at variable $i \in \mathcal{V}$ consists of all paths on the factor graph, starting at i, that never backtrack (see also Figure 9(c)). This means that the bounds on the (exact) marginals that we just derived are at the same time bounds on the approximate Belief Propagation marginals (beliefs).

Corollary 11 *In the situation described in Theorem 10, the final bounding box $\mathcal{B}_i$ also bounds the (approximate) Belief Propagation marginal of the root node i, i.e., $\mathbb{P}_{BP}(x_i) \in \mathcal{B}_i$.* ■

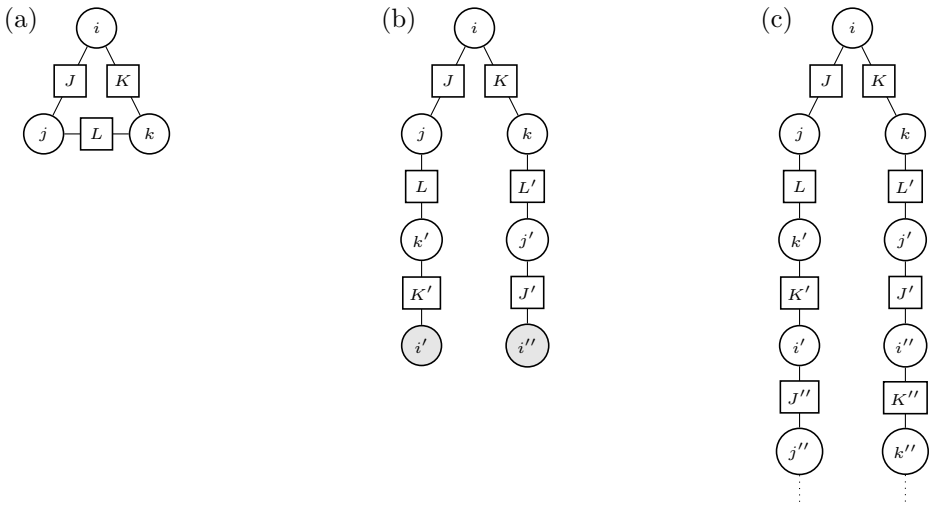


Figure 9: (a) Factor graph; (b) Self-avoiding walk tree with root i, with cycle-induced leaf nodes shown in gray; (c) Computation tree for i.

2.9 Bounds using Self-Avoiding Walk Trees

While writing this article, we became aware that a related method to obtain bounds on single-variable marginals has been proposed recently by Ihler (2007).[4] The method presented there uses a different local bound, which empirically seems to be less tight than ours, but has the advantage of being computationally less demanding if the domains of the random variables are large. On the other hand, the bound presented there does not use subtrees of the factor graph, but uses self-avoiding walk (SAW) trees instead. Since each subtree of the factor graph is a subtree of an SAW tree, this may lead to tighter bounds.

The idea of using a self-avoiding walk tree for calculating marginal probabilities seems to be generally attributed to Weitz (2006), but can already be found in (Scott and Sokal, 2005). In this subsection, we show how this idea can be combined with the propagation of bounding boxes. The result Theorem 13 will turn out to be an improvement over Theorem 10 in case there are only pairwise interactions, whereas in the general case, Theorem 10 often yields tighter bounds empirically.

Definition 12 *Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph and let $i \in \mathcal{V}$. A* self-avoiding walk (SAW) *starting at $i \in \mathcal{V}$ of length $n \in \mathbb{N}^*$ is a sequence $\alpha = (\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n) \in (\mathcal{V} \cup \mathcal{F})^n$ that*

(i) starts at $i \in \mathcal{V}$, i.e., $\alpha_1 = i$;

4. Note that (Ihler, 2007, Lemma 5) contains an error: to obtain the correct expression, one has to replace δ with δ^2, i.e., the correct statement would be that

$$\frac{m(j)}{\delta^2 + (1-\delta^2)m(j)} \leq p(j) \leq \frac{\delta^2 m(j)}{1 - (1-\delta^2)m(j)}$$

if $d\big(p(x)/m(x)\big) \leq \delta$ (where p and m should both be normalized).

(ii) subsequently visits neighboring nodes in the factor graph, i.e., $\alpha_{j+1} \in N_{\alpha_j}$ *for all* $j = 1, 2, \ldots, n-1$*;*

(iii) does not backtrack, i.e., $\alpha_j \neq \alpha_{j+2}$ *for all* $j = 1, 2, \ldots, n-2$*;*

(iv) the first $n-1$ *nodes are all different, i.e.,* $\alpha_j \neq \alpha_k$ *if* $j \neq k$ *for* $j, k \in \{1, 2, \ldots, n-1\}$.[5]

The set of all self-avoiding walks starting at $i \in \mathcal{V}$ *has a natural tree structure, defined by declaring each SAW* $(\alpha_1, \alpha_2, \ldots, \alpha_n, \alpha_{n+1})$ *to be a child of the SAW* $(\alpha_1, \alpha_2, \ldots, \alpha_n)$*, for all* $n \in \mathbb{N}^*$*; the resulting tree is called the* self-avoiding walk (SAW) tree with root $i \in \mathcal{V}$*, denoted* $T_{\mathcal{G}}^{SAW}(i)$*.*

Note that the name "self-avoiding walk tree" is slightly inaccurate, because the last node of a SAW may have been visited already. In general, the SAW tree can be much larger than the original factor graph. Following Ihler (2007), we call a leaf node in the SAW tree a *cycle-induced leaf node* if it contains a cycle (i.e., if its final node has been visited before in the same walk), and call it a *dead-end leaf node* otherwise. We denote the parent of node α in the SAW tree by $\mathrm{par}(\alpha)$ and we denote its children by $\mathrm{ch}(\alpha)$. The final node of a SAW $\alpha = (\alpha_1, \ldots, \alpha_n)$ is denoted by $\mathcal{G}(\alpha) = \alpha_n$. An example of a SAW tree for our running example factor graph is shown in Figure 9(b).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph and let $i \in \mathcal{V}$. We now define a propagation algorithm on the SAW tree $T_{\mathcal{G}}^{SAW}(i)$, where each node $\alpha \in T_{\mathcal{G}}^{SAW}(i)$ (except for the root i) sends a message $\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)}$ to its parent node $\mathrm{par}(\alpha) \in T_{\mathcal{G}}^{SAW}(i)$. Each cycle-induced leaf node of $T_{\mathcal{G}}^{SAW}(i)$ sends a simplex to its parent node: if α is a cycle-induced leaf node, then

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \begin{cases} \mathcal{P}_{\mathcal{G}(\alpha)} & \text{if } \mathcal{G}(\alpha) \in \mathcal{V} \\ \mathcal{P}_{\mathcal{G}(\mathrm{par}(\alpha))} & \text{if } \mathcal{G}(\alpha) \in \mathcal{F}. \end{cases} \tag{3}$$

All other nodes α in the SAW tree (i.e., the dead-end leaf nodes and the nodes with children, except for the root i) send a message according to the following rules. If $\mathcal{G}(\alpha) \in \mathcal{V}$,

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \begin{cases} \displaystyle\prod_{\beta \in \mathrm{ch}(\alpha)} \mathcal{B}_{\beta \to \alpha} & \text{if all } \mathcal{B}_{\beta \to \alpha} \text{ are boxes} \\ \mathcal{P}_{\mathcal{G}(\alpha)} & \text{if at least one of the } \mathcal{B}_{\beta \to \alpha} \text{ is a simplex.} \end{cases} \tag{4}$$

On the other hand, if $\mathcal{G}(\alpha) \in \mathcal{F}$,

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \mathcal{BN}\left(\sum_{x_{\setminus \mathcal{G}(\mathrm{par}(\alpha))}} \psi_{\mathcal{G}(\alpha)} \mathcal{B}\left(\prod_{\beta \in \mathrm{ch}(\alpha)} \mathcal{B}_{\beta \to \alpha} \right) \right). \tag{5}$$

The final "belief" at the root node $i \in \mathcal{V}$ is defined as:

$$\mathcal{B}_i = \begin{cases} \mathcal{BN}\left(\displaystyle\prod_{\beta \in \mathrm{ch}(i)} \mathcal{B}_{\beta \to i} \right) & \text{if all } \mathcal{B}_{\beta \to i} \text{ are boxes} \\ \mathcal{P}_{\mathcal{G}(i)} & \text{if at least one of the } \mathcal{B}_{\beta \to i} \text{ is a simplex.} \end{cases} \tag{6}$$

5. Note that (iii) almost follows from (iv), except for the condition that $\alpha_{n-2} \neq \alpha_n$.

We will refer to this algorithm as "box propagation on the SAW tree"; it is similar to the propagation algorithm for boxes on subtrees of the factor graph that we defined earlier. However, note that whereas (2) bounds a sum-product assuming that incoming measures factorize, (5) is a looser bound that also holds if the incoming measures do not necessarily factorize. In the special case where the factor depends only on two variables, the updates (2) and (5) are identical.

Theorem 13 *Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ and let $T_{\mathcal{G}}^{SAW}(i)$ be the SAW tree with root i. Then $\mathbb{P}(x_i) \in \mathcal{B}_i$, where $\mathcal{B}_i$ is the bounding box that results from propagating bounds on the SAW tree $T_{\mathcal{G}}^{SAW}(i)$ according to equations (3)–(6).*

The following lemma, illustrated in Figure 10, plays a crucial role in the proof of the theorem. It seems to be related to the so-called "telegraph expansion" used in Weitz (2006).

Lemma 14 *Let $A, C \subseteq \mathcal{V}$ be two disjoint sets of variable indices and let $\Psi \in \mathcal{M}_{A \cup C}$ be a factor depending on (some of) the variables in $A \cup C$. Then:*

$$\mathcal{N}\left(\sum_{x_C} \Psi\right) \in \mathcal{B}\left(\prod_{i \in A} B_i\right)$$

where

$$B_i := \mathcal{B}\mathcal{N}\left(\sum_{x_{A \setminus i}} \sum_{x_C} \Psi \mathcal{Q}^*_{A \setminus i}\right).$$

Proof We assume that $C = \emptyset$; the more general case then follows from this special case by replacing Ψ by $\sum_{x_C} \Psi$.

Let $A = \{i_1, i_2, \ldots, i_n\}$ and let $\underline{\Psi}_i$, $\overline{\Psi}_i$ be the lower and upper bounds corresponding to B_i, for all $i \in A$. For each $k = 1, 2, \ldots, n$, note that

$$\left(\prod_{l=1}^{k-1} \mathbf{1}_{i_l}\right)\left(\prod_{l=k+1}^{n} \delta_{x_{i_l}}\right) \in \mathcal{Q}^*_{A \setminus i_k},$$

for all $x_{\{i_{k+1}, \ldots, i_n\}} \in \mathcal{X}_{\{i_{k+1}, \ldots, i_n\}}$. Therefore, we obtain from the definition of B_{i_k} that

$$\forall x_A \in \mathcal{X}_A : \qquad \underline{\Psi}_{i_k} \quad \leq \quad \frac{\sum_{x_{i_{k-1}}} \cdots \sum_{x_{i_1}} \Psi}{\sum_{x_{i_k}} \sum_{x_{i_{k-1}}} \cdots \sum_{x_{i_1}} \Psi} \quad \leq \quad \overline{\Psi}_{i_k}$$

for all $k = 1, 2, \ldots, n$. Taking the product of these n inequalities yields

$$\prod_{k=1}^{n} \underline{\Psi}_{i_k} \leq \mathcal{N}\Psi \leq \prod_{k=1}^{n} \overline{\Psi}_{i_k}$$

pointwise, and therefore $\mathcal{N}\Psi \in \mathcal{B}\left(\prod_{k=1}^{n} B_{i_k}\right)$. ■

The following corollary is somewhat elaborate to state, but readily follows from the previous lemma after attaching a factor I that depends on all nodes in A and one additional newly introduced node i:

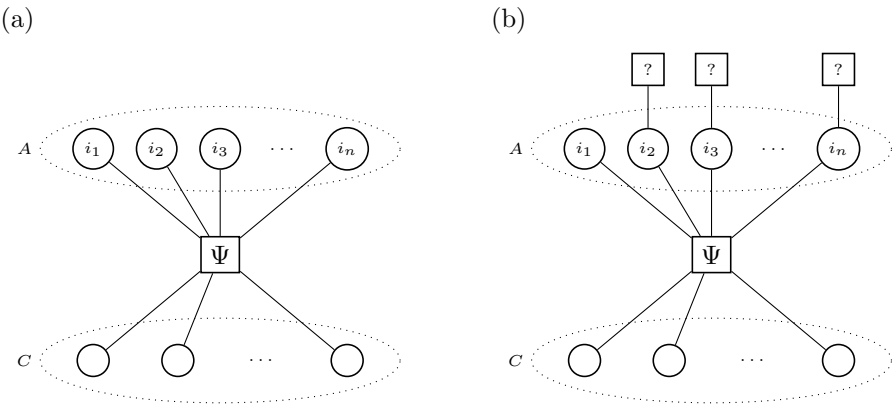


Figure 10: The second basic lemma: the marginal on x_A in (a) is contained in the bounding box of the product of smallest bounding boxes B_i for $i \in A$, where (b) the smallest bounding box B_i is obtained by putting arbitrary factors on the other variables in $A \setminus \{i\}$ and calculating the smallest bounding box on i, illustrated here for the case $i = i_1$.

Corollary 15 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ with exactly one neighbor in $\mathcal{F}$, say $N_i = \{I\}$. Then $\mathbb{P}(x_i) \in B_i$ where*

$$B_i = \mathcal{BN}\left(\sum_{x_{N_I \setminus i}} \psi_I \mathcal{B}\left(\prod_{k \in N_I \setminus i} B_k^{\setminus I}\right)\right) \tag{7}$$

and

$$B_k^{\setminus I} = \mathcal{BN}\left(\sum_{x_{\mathcal{V} \setminus \{i,k\}}} \Psi_{\mathcal{F} \setminus I} \mathcal{Q}^*_{N_I \setminus \{i,k\}}\right)$$

with

$$\Psi_{\mathcal{F} \setminus I} := \prod_{J \in \mathcal{F} \setminus I} \psi_J.$$

■

We now proceed with a sketch of the proof of Theorem 13, which was inspired by (Ihler, 2007).

Proof sketch The proof proceeds using structural induction, recursively transforming the original factor graph $\mathcal{G}$ into the SAW tree $T_{\mathcal{G}}^{SAW}(i)$, refining the bound at each step, until it becomes equivalent to the result of the message propagation algorithm on the SAW tree described above in equations (3)–(6).

Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ and let $T_{\mathcal{G}}^{SAW}(i)$ be the SAW tree with root i. Let $\{I_1, \ldots, I_n\} = N_i$.

Suppose that $n > 1$. Consider the equivalent factor graph $\mathcal{G}'$ that is obtained by creating n copies i_n of the variable node i, where each copy i_j is only connected with the factor node I_j (for $j = 1, \ldots, n$); in addition, all copies are connected with the original variable i using

the delta function $\psi_\delta := \delta(x_i, x_{i_1}, \ldots, x_{i_n})$. This step is illustrated in Figure 11(a)–(b). Applying Corollary 15 to $\mathcal{G}'$ yields the following bound which follows from (7) because of the properties of the delta function:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\prod_{j=1}^{n} B_{i_j}^{\backslash\delta}\right) \tag{8}$$

where

$$B_{i_j}^{\backslash\delta} := \mathcal{BN}\left(\sum_{x_{\backslash i_j}}\left(\prod_{J\in\mathcal{F}} \psi_J\right) \mathcal{Q}^*_{\{i_1,\ldots,i_{j-1},i_{j+1},\ldots,i_n\}}\right) \qquad j = 1, \ldots, n.$$

In the expression on the right-hand side, the factor ψ_{I_k} implicitly depends on i_k instead of i (for all $k = 1, \ldots, n$). This bound is represented graphically in Figure 11(c)–(d) where the gray variable nodes correspond to simplices of single-variable factors, i.e., they are meant to be multiplied with unknown single-variable factors. Note that (8) corresponds precisely with (6).

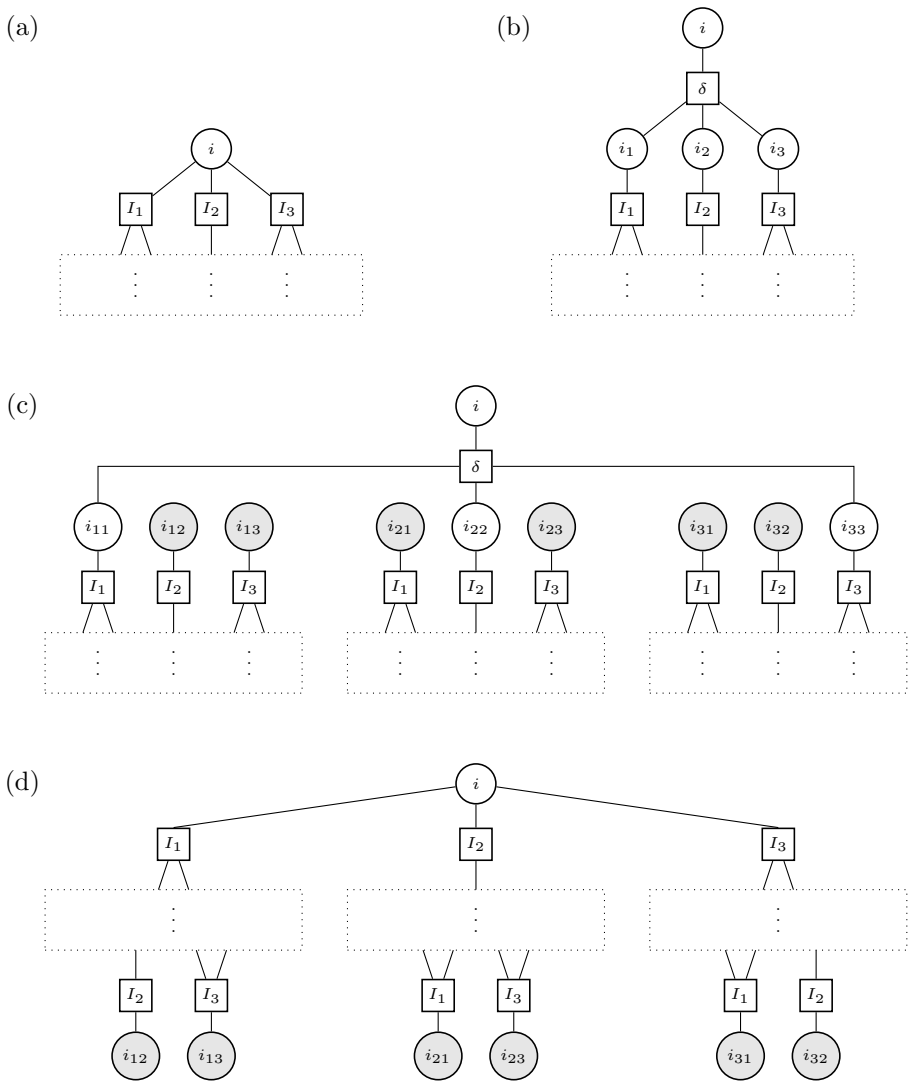


Figure 11: One step in the proof of Theorem 13: propagating bounds towards variable i in case it has more than one neighboring factor nodes $I_1, \ldots, I_n$ (here, $n = 3$). Gray nodes represent added (unknown) single-variable factors. (a) Factor graph $\mathcal{G}$. (b) Equivalent factor graph $\mathcal{G}'$. (c) Result of replicating $\mathcal{G}$ n times, where in each copy $\mathcal{G}_k$ of $\mathcal{G}$, i is replaced by exactly n copies i_{kj} of i for $j = 1, \ldots, n$, where i_{kj} is connected only with the factor I_j in $\mathcal{G}_k$. Then, the original variable i is connected using a delta factor with n of its copies i_{jj} for $j = 1, \ldots, n$. (d) Simplification of (c) obtained by identifying i with its n copies i_{jj} for $j = 1, \ldots, n$ and changing the layout slightly.

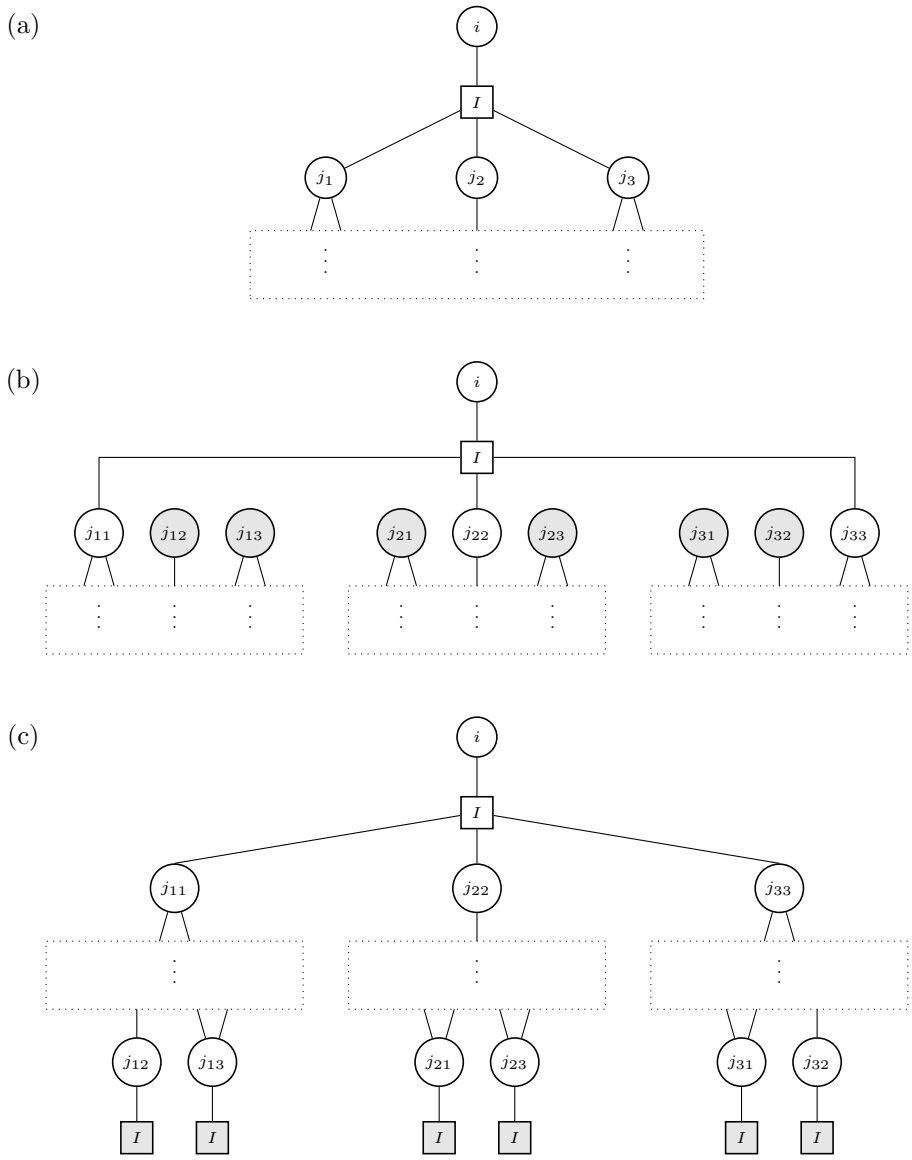


Figure 12: Another step in the proof of Theorem 13: propagating bounds towards variable i in case it has exactly one neighboring factor node I which has $m+1$ neighboring variables $\{i, j_1, \ldots, j_m\}$. (a) Factor graph $\mathcal{G}$. (b) Result of replicating $\mathcal{G} \setminus \{i, I\}$ m times and connecting the factor I with i and with copy j_{kk} of j_k for $k = 1, \ldots, m$. (c) Equivalent to (b) but with a slightly changed layout. The gray copies of I represent (unknown) single-variable factors (on their neighboring variable).

The case that $n = 1$ is simpler because there is no need to introduce the delta function. It is illustrated in Figure 12. Let $\{I\} = N_i$ and let $\{j_1, \ldots, j_m\} = N_I \setminus i$. Applying Corollary 15 yields the following bound:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\sum_{x_{N_I \setminus i}} \psi_I \mathcal{B}\left(\prod_{k=1}^{m} B_{j_k}^{\setminus I}\right)\right) \tag{9}$$

where

$$B_{j_k}^{\setminus I} := \mathcal{BN}\left(\sum_{x_{\setminus\{i,j_k\}}} \left(\prod_{J \in \mathcal{F} \setminus I} \psi_J\right) \mathcal{Q}^*_{\{j_1,\ldots,j_{k-1},j_{k+1},\ldots,j_m\}}\right) \qquad k = 1, \ldots, m.$$

This bound is represented graphically in Figure 12(b)–(c) where the gray nodes correspond with simplices of single-variable factors. Note that (9) corresponds precisely with (5).

Recursively iterating the factor graph operations in Figures 12 and 11, the connected component that remains in the end is precisely the SAW tree $T_{\mathcal{G}}^{SAW}(i)$; the bounds derived along the way correspond precisely with the message passing algorithm on the SAW tree described above. ■

Again, the self-avoiding walk tree with root i is a subtree of the computation tree for i. This means that the bounds on the exact marginals given by Theorem 13 are bounds on the approximate Belief Propagation marginals (beliefs) as well.

Corollary 16 *In the situation described in Theorem 13, the final bounding box $\mathcal{B}_i$ also bounds the (approximate) Belief Propagation marginal of the root node i, i.e., $\mathbb{P}_{BP}(x_i) \in \mathcal{B}_i$.* ■

3. Related work

There exist many other bounds on single-variable marginals. Also, bounds on the partition sum can be used to obtain bounds on single-variable marginals. For all bounds known to the authors, we will discuss how they compare with our bounds. In the following, we will denote exact marginals as $p_i(x_i) := \mathbb{P}(x_i)$ and BP marginals (beliefs) as $b_i(x_i) := \mathbb{P}_{BP}(x_i)$.

3.1 The Dobrushin-Tatikonda bound

Tatikonda (2003) derived a bound on the error of BP marginals using mathematical tools from Gibbs measure theory (Georgii, 1988), in particular using a result known as Dobrushin's theorem. The bounds on the error of the BP marginals can be easily translated into bounds on the exact marginals:

$$|b_i(x_i) - p_i(x_i)| \leq \epsilon \implies p_i(x_i) \in [b_i(x_i) - \epsilon, b_i(x_i) + \epsilon]$$

for all $i \in \mathcal{V}$ and $x_i \in \mathcal{X}_i$.

The Dobrushin-Tatikonda bound depends on the *girth* of the graph (the number of edges in the shortest cycle, or infinity if there is no cycle) and the properties of Dobrushin's

interdependence matrix, which is a $N \times N$ matrix C. The entry C_{ij} is only nonzero if $i \in \partial j$ and in that case, the computational cost of computing its value is exponential in the size of the Markov blanket. Thus the computational complexity of the Dobrushin-Tatikonda bound is $\mathcal{O}(\max_{i \in \mathcal{V}} \#(\mathcal{X}_{\partial i}))$, plus the cost of running BP.

3.2 The Dobrushin-Taga-Mase bound

Inspired by the work of Tatikonda and Jordan (2002), Taga and Mase (2006) derived another bound on the error of BP marginals, also based on Dobrushin's theorem. This bound also depends on the properties of Dobrushin's interdependence matrix and has similar computational cost. Whereas the Dobrushin-Tatikonda bound gives one bound for all variables, the Dobrushin-Taga-Mase bound gives a different bound for each variable.

3.3 Bound Propagation

Leisink and Kappen (2003) proposed a method called "Bound Propagation" which can be used to obtain bounds on exact marginals. The idea underlying this method is very similar to the one employed in this work, with one crucial difference. Whereas we use a cavity approach, using as basis equation

$$\mathbb{P}(x_i) \propto \sum_{x_{\partial i}} \left(\prod_{I \in N_i} \psi_I \right) \mathbb{P}^{\backslash i}(x_{\partial i}), \qquad \mathbb{P}^{\backslash i}(x_{\partial i}) \propto \sum_{x_{\mathcal{V} \backslash \Delta i}} \prod_{I \in \mathcal{F} \backslash N_i} \psi_I$$

and bound the quantity $\mathbb{P}(x_i)$ by optimizing over $\mathbb{P}^{\backslash i}(x_{\partial i})$, the basis equation employed by Bound Propagation is

$$\mathbb{P}(x_i) = \sum_{x_{\partial i}} \mathbb{P}(x_i \,|\, x_{\partial i}) \mathbb{P}(x_{\partial i})$$

and the optimization is over $\mathbb{P}(x_{\partial i})$. Unlike in our case, the computational complexity is exponential in the size of the Markov blanket, because of the required calculation of the conditional distribution $\mathbb{P}(x_i \,|\, x_{\partial i})$. On the other hand, the advantage of this approach is that a bound on $\mathbb{P}(x_j)$ for $j \in \partial i$ is also a bound on $\mathbb{P}(x_{\partial i})$, which in turn gives rise to a bound on $\mathbb{P}(x_i)$. In this way, bounds can propagate through the graphical model, eventually yielding a new (tighter) bound on $\mathbb{P}(x_{\partial i})$. Although the iteration can result in rather tight bounds, the main disadvantage of Bound Propagation is its computational cost: it is exponential in the Markov blanket and often many iterations are needed for the bounds to become tight. Indeed, for a simple tree of $N = 100$ variables, it can happen that Bound Propagation needs several minutes and still obtains very loose bounds (whereas our bounds give the exact marginal as lower and as upper bound, i.e., they arrive at the optimally tight bound).

3.4 Upper and lower bounds on the partition sum

Various upper and lower bounds on the partition sum Z in (1) exist. An upper and a lower bound on Z can be combined to obtain bounds on marginals in the following way. First, note that the exact marginal of i satisfies

$$\mathbb{P}(x_i) = \frac{Z_i(x_i)}{Z},$$

where we defined the partition sum of the *clamped* model as follows:

$$Z_i(x_i) := \sum_{x_{\mathcal{V}\setminus\{i\}}} \prod_{I\in\mathcal{F}} \psi_I.$$

Thus, we can bound

$$\frac{Z_i^-(x_i)}{Z^+} \leq p_i(x_i) \leq \frac{Z_i^+(x_i)}{Z^-}$$

where $Z^- \leq Z \leq Z^+$ and $Z_i^-(x_i) \leq Z_i(x_i) \leq Z_i^+(x_i)$ for all $x_i \in \mathcal{X}_i$.

A well-known lower bound of the partition sum is the Mean Field bound. A tighter lower bound was derived by Leisink and Kappen (2001). An upper bound on the log partition sum was derived by Wainwright et al. (2005). Other lower and upper bounds (for the case of binary variables with pairwise interactions) have been derived by Jaakkola and Jordan (1996).

4. Experiments

We have done several experiments to compare the quality and computation time of various bounds empirically. For each variable in the factor graph under consideration, we calculated the *gap* for each bound $\mathcal{B}_i\left(\underline{\Psi}_i, \overline{\Psi}_i\right) \ni \mathbb{P}(x_i)$, which we define as the ℓ_0-norm $\left\|\overline{\Psi}_i - \underline{\Psi}_i\right\|_0 = \max_{x_i\in\mathcal{X}_i} \left|\overline{\Psi}_i(x_i) - \underline{\Psi}_i(x_i)\right|$.

We have used the following bounds in our comparison:

DT**:** Dobrushin-Tatikonda (Tatikonda, 2003, Proposition V.6).

DTM**:** Dobrushin-Taga-Mase (Taga and Mase, 2006, Theorem 1).

BoundProp**:** Bound Propagation (Leisink and Kappen, 2003), using the implementation of M. Leisink, where we chose the maximum cluster size to be $\max_{i\in\mathcal{V}} \#(\Delta i)$.

BoxProp-SubT**:** Theorem 10, where we used a simple breadth-first algorithm to recursively construct the subtree.

BoxProp-SAWT**:** Theorem 13, where we truncated the SAW tree to at most 5000 nodes.

Ihler-SAWT**:** Ihler's bound (Ihler, 2007). This bound has only been formulated for pairwise interactions.

Ihler-SubT**:** Ihler's bound (Ihler, 2007) applied on a truncated version of the SAW tree, namely on the same subtree as used in BoxProp-SubT. This bound has only been formulated for pairwise interactions.

In addition, we compared with appropriate combinations of the following bounds:

MF**:** Mean-field lower bound.

LK3**:** Third-order lower bound (Leisink and Kappen, 2001, Eq. (10)), where we took for μ_i the mean field solutions. This bound has been formulated only for the binary, pairwise case.

JJ: Refined upper bound (Jaakkola and Jordan, 1996, Section 2.2), with a greedy optimization over the parameters. This bound has been formulated only for the binary, pairwise case.

TRW: Our implementation of (Wainwright et al., 2005). This bound has been formulated only for pairwise interactions.

For reference, we calculated the Belief Propagation (BP) errors by comparing with the exact marginals, using the ℓ_0 distance as error measure.

4.1 Grids with binary variables

We considered a 5×5 Ising grid with binary (± 1-valued) variables, i.i.d. spin-glass nearest-neighbor interactions $J_{ij} \sim \mathcal{N}(0, \beta^2)$ and i.i.d. local fields $\theta_i \sim \mathcal{N}(0, \beta^2)$, with probability distribution

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

We took one random instance of the parameters J and θ (drawn for $\beta = 1$) and scaled these parameters with the interaction strength parameter β, for which we took values in $\{10^{-2}, 10^{-1}, 1, 10\}$.

The results are shown in Figure 13. For very weak interactions ($\beta = 10^{-2}$), BoxProp-SAWT gave the tightest bounds of all other methods, the only exception being BoundProp, which gave a somewhat tighter bound for 5 variables out of 25. For weak and moderate interactions ($\beta = 10^{-1}, 1$), BoxProp-SAWT gave the tightest bound of all methods for each variable. For strong interactions ($\beta = 10$), the results were mixed, the best methods being BoxProp-SAWT, BoundProp, MF-TRW and LK3-TRW. Of these, BoxProp-SAWT was the fastest method, whereas the methods using TRW were the slowest.[6] For $\beta = 10$, we present scatter plots in Figure 14 to compare the results of some methods in more detail. These plots illustrate that the tightness of bounds can vary widely over methods and variables.

Among the methods yielding the tightest bounds, the computation time of our bounds is relatively low in general; only for low interaction strengths BoundProp is faster than BoxProp-SAWT. Furthermore, the computation time of our bounds does not depend on the interaction strength, in contrast with iterative methods such as BoundProp and MF-TRW, which need more iterations for increasing interaction strength (as the variables become more and more correlated). Further, as expected, BoxProp-SubT needs less computation time than BoxProp-SAWT but also yields less tight bounds. Another observation is that our bounds outperform the related versions of Ihler's bounds.

6. We had to loosen the convergence criterion for the inner loop of TRW, otherwise it would have taken hours. Since some of the bounds are significantly tighter than the convergence criterion we used, this may suggest that one can loosen the convergence criterion for TRW even more and still obtain good results using less computation time than the results we present here. Unfortunately, it is not clear how this criterion should be chosen in an optimal way without actually trying different values and using the best one.

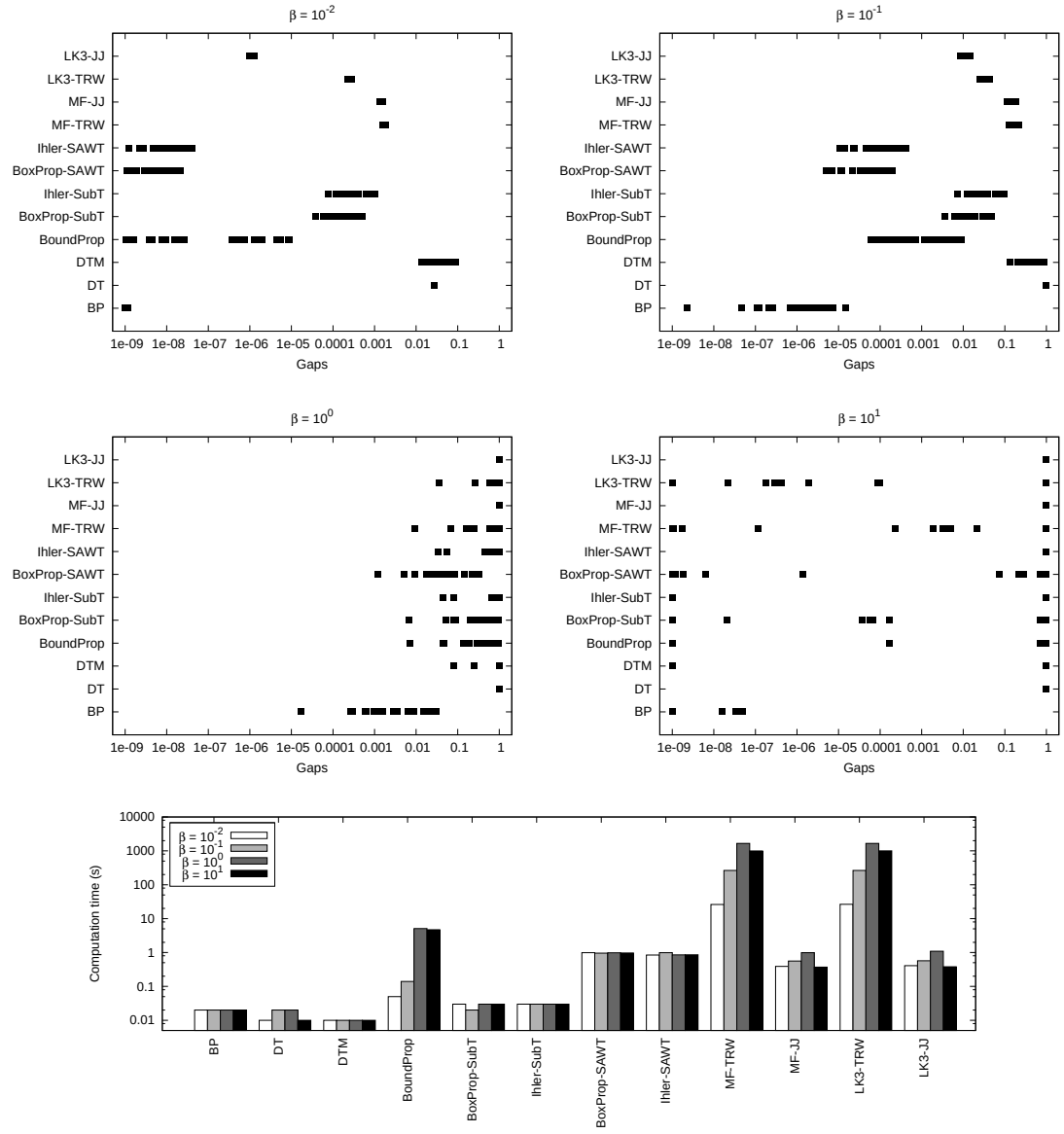


Figure 13: Results for grids with binary variables. The first four graphs show, for different values of the interaction strength β, the gaps of bounds on marginals calculated using different methods. Also shown for reference are the errors in the BP approximations to the same marginals. The final graph shows the total computation time for each method.

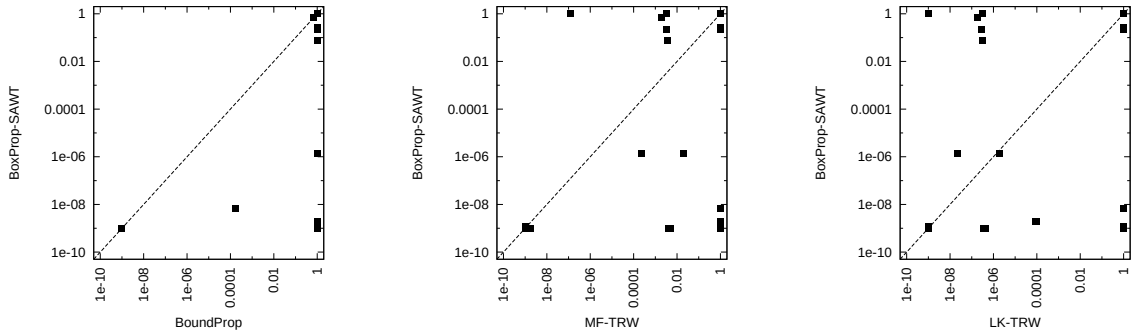


Figure 14: Scatter plots comparing some methods in detail for grids with binary variables for strong interactions ($\beta = 10$).

4.2 Grids with ternary variables

To evaluate the bounds beyond the special case of binary variables, we have performed experiments on a 5×5 grid with ternary variables and pairwise factors between nearest-neighbor variables on the grid. The entries of the factors were i.i.d., drawn by taking a random number from a normal distribution $\mathcal{N}\left(0, \beta^2\right)$ with mean 0 and standard deviation β and taking the exp of that random number.

The results are shown in Figure 15. We have not compared with bounds involving JJ or LK3 because these methods have only been formulated originally for the case of binary variables. This time, our method BoxProp-SAWT yielded the tightest bounds for all interaction strengths and for all variables (although this is not immediately clear from the plots).

4.3 Medical diagnosis

We also applied the bounds on simulated Promedas patient cases (Wemmenhove et al., 2007). These factor graphs have binary variables and singleton, pairwise and triple interactions (containing zeros). Two examples of such factor graphs are shown in Figure 16. Because of the triple interactions, less methods were available for comparison.

The results of various bounds for nine different, randomly generated, instances are shown in Figure 17. The total number of variables for these nine instances was 1270. The total computation time needed for BoxProp-SubT was 51 s, for BoxProp-SAWT 149 s, for BoundProp more than 75000 s (we aborted the method for two instances because convergence was very slow, which explains the missing results in the plot) and to calculate the Belief Propagation errors took 254 s. BoundProp gave the tightest bound for only 1 out of 1270 variables, BoxProp-SAWT for 5 out of 1270 variables and BoxProp-SubT gave the tightest bound for the other 1264 variables.

Interestingly, whereas for pairwise interactions, BoxProp-SAWT gives tighter bounds than BoxProp-SubT, for the factor graphs considered here, the bounds calculated by BoxProp-SAWT were generally less tight than those calculated by BoxProp-SubT. This

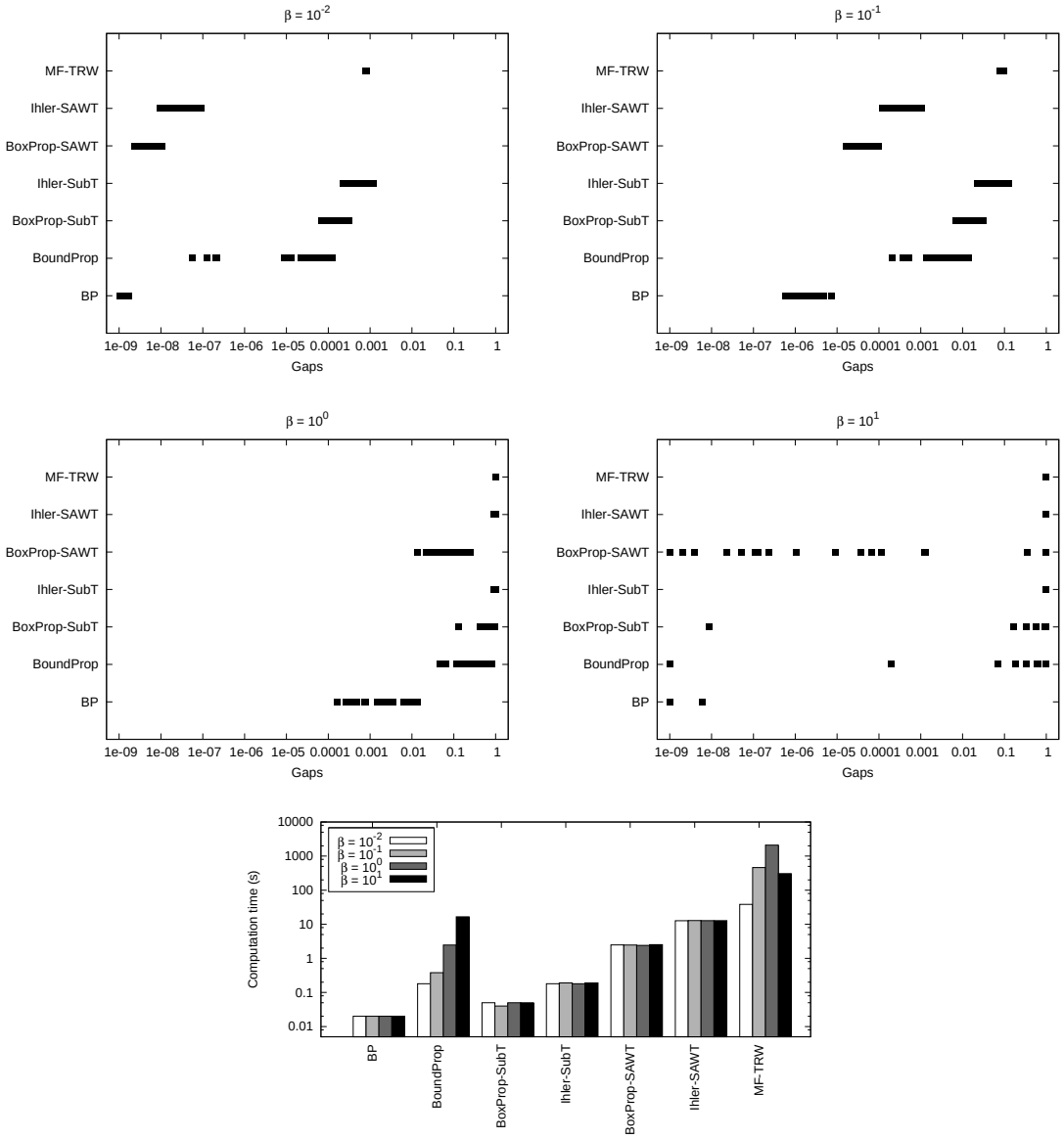


Figure 15: Results for grids with ternary variables. The first four graphs show, for different values of the interaction strength β, the gaps of bounds on marginals calculated using different methods. Also shown for reference are the errors in the BP approximations to the same marginals. The final graph shows the total computation time for each method.

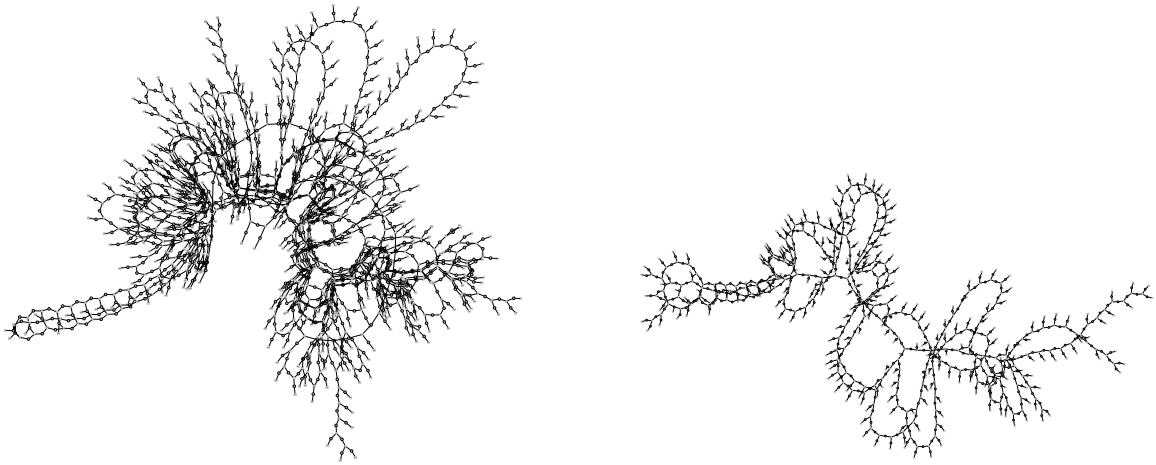


Figure 16: Two of the Promedas factor graphs.

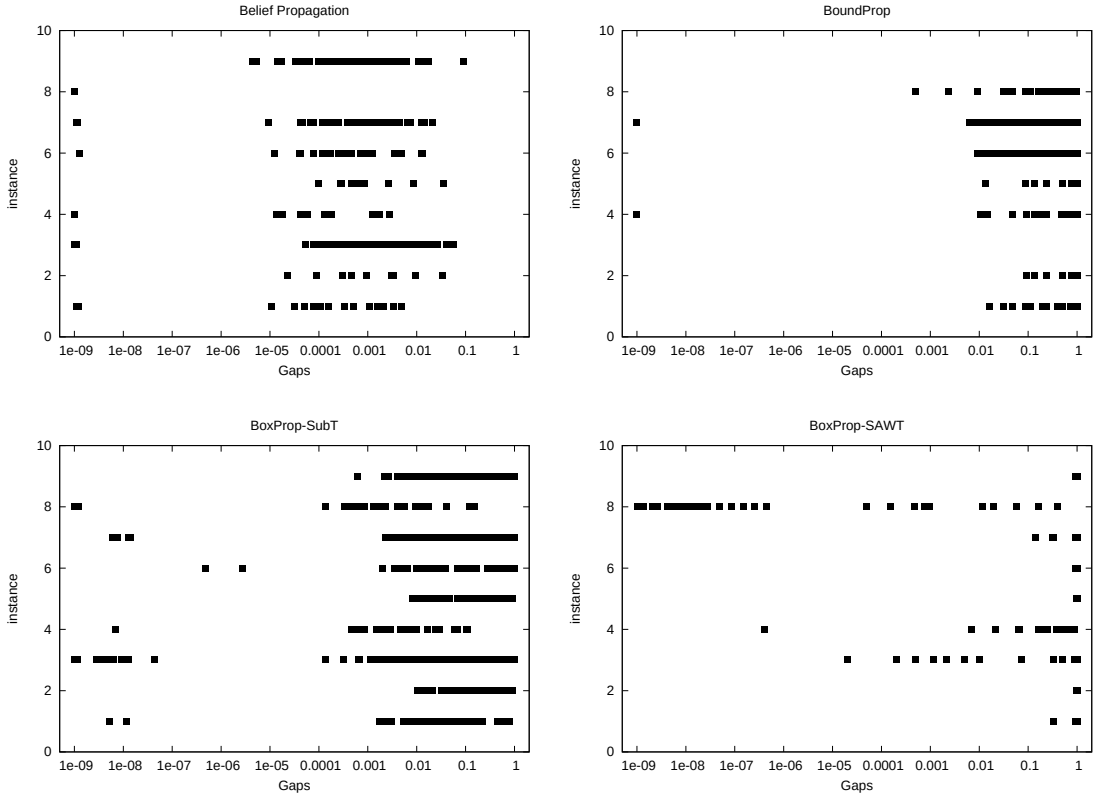


Figure 17: Results for nine different factor graphs corresponding to simulated Promedas patient cases. In reading order: Belief Propagation errors, BoundProp, BoxProp-SubT and BoxProp-SAWT.

is presumably due to the local bound (5) needed on the SAW tree, which is quite loose compared with the local bound (2) that assumes independent incoming bounds.

Not only does BoxProp-SubT give the tightest bounds for almost all variables, it is also the fastest method. Finally, note that the tightness of these bounds still varies widely depending on the instance and on the variable of interest.

5. Conclusion and discussion

We have derived two related novel bounds on exact single-variable marginals. Both bounds also bound the approximate Belief Propagation marginals. The bounds are calculated by propagating convex sets of measures over a subtree of the computation tree, with update equations resembling those of BP. For variables with a limited number of possible values, the bounds can be computed efficiently. Empirically, our bounds often outperform the existing state-of-the-art in that case. Although we have only shown results for factor graphs for which exact inference was still tractable (in order to be able to calculate the BP error), we would like to stress here that it is not difficult to construct factor graphs for which exact inference is no longer tractable but the bounds can still be calculated efficiently. An example are large Ising grids (of size $m \times m$ with m larger than 30). Indeed, for binary Ising grids, the computation time of the bounds (for all variables in the network) scales linearly in the number of variables, assuming that we truncate the subtrees and SAW trees to a fixed maximum size.

Whereas the results of different approximate inference methods usually cannot be combined in order to get a better estimate of marginal probabilities, for bounds one can combine different methods simply by taking the tightest bound or the intersection of the bounds. Thus it is generally a good thing to have different bounds with different properties (such as tightness and computation time).

An advantage of our methods BoxProp-SubT and BoxProp-SAWT over iterative methods like BoundProp and MF-TRW is that the computation time of the iterative methods is difficult to predict (since it depends on the number of iterations needed to converge which is generally not known a priori). In contrast, the computation time needed for our bounds BoxProp-SubT and BoxProp-SAWT only depends on the structure of the factor graph (and the chosen subtree) and is independent of the values of the interactions. Furthermore, by truncating the tree one can trade some tightness for computation time.

By far the slowest methods turned out to be those combining the upper bound TRW with a lower bound on the partition sum. The problem here is that TRW usually needs many iterations to converge, especially for stronger interactions where convergence rate can go down significantly. In order to prevent exceedingly long computations, we had to hand-tune the convergence criterion of TRW according to the case at hand.

BoundProp can compete in certain cases with the bounds derived here, but more often than not it turned out to be rather slow or did not yield very tight bounds. Although BoundProp also propagates bounding boxes over measures, it does this in a slightly different way which does not exploit independence as much as our bounds. On the other hand, it can propagate bounding boxes several times, refining the bounds more and more each iteration.

Regarding the related bounds BoxProp-SubT, BoxProp-SAWT and Ihler-SAWT we can draw the following conclusions. For pairwise interactions and variables that have not too many possible values, BoxProp-SAWT is the method of choice, yielding the tightest bounds without needing too much computation time. The bounds are more accurate than the bounds produced by Ihler-SAWT due to the more precise local bound that is used; the difference is largest for strong interactions. However, the computation time of this more precise local bound is exponential in the number of possible values of the variables, whereas the local bound used in Ihler-SAWT is only polynomial in the number of possible values of the variables. Therefore, if this number is large, BoxProp-SAWT may be no longer applicable in practice, whereas Ihler-SAWT still may be applicable. If factors are present that depend on more than two variables, it seems that BoxProp-SubT is the best method to obtain tight bounds, especially if the interactions are strong. Note that it is not immediately obvious how to extend Ihler-SAWT beyond pairwise interactions, so we could not compare with that method in that case.

This work also raises some new questions and opportunities for future work. First, the bounds can be used to generalize the improved conditions for convergence of Belief Propagation that were derived in (Mooij and Kappen, 2007) beyond the special case of binary variables with pairwise interactions. Second, it may be possible to combine the various ingredients in BoundProp, BoxProp-SubT and BoxProp-SAWT in novel ways in order to obtain even better bounds. Third, it is an interesting open question whether the bounds can be extended to continuous variables in some way. Finally, although our bounds are a step forward in quantifying the error of Belief Propagation, the actual error made by BP is often at least one order of magnitude lower than the tightness of these bounds. This is due to the fact that (loopy) BP cycles information through loops in the factor graph; this cycling apparently improves the results. The interesting and still unanswered question is why it makes sense to cycle information in this way and whether this error reduction effect can be quantified.

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) and was also sponsored in part by the Dutch Technology Foundation (STW).

We thank Wim Wiegerinck for several fruitful discussions, Bastian Wemmenhove for providing the Promedas test cases, and Martijn Leisink for kindly providing his implementation of Bound Propagation.

References

G.F. Cooper. The computational complexity of probabistic inferences. *Artificial Intelligence*, 42(2-3):393–405, March 1990.

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

A. Ihler. Accuracy bounds for belief propagation. In *Proceedings of the 23th Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, July 2007.

T. S. Jaakkola and M. Jordan. Recursive algorithms for approximating probabilities in graphical models. In *Proc. Conf. Neural Information Processing Systems (NIPS 9)*, pages 487–493, Denver, CO, 1996.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

M. Leisink and B. Kappen. Bound propagation. *Journal of Artificial Intelligence Research*, 19:139–154, 2003.

M. A. R. Leisink and H. J. Kappen. A tighter bound for graphical models. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 13 (NIPS*2000)*, pages 266–272, Cambridge, MA, 2001. MIT Press.

J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437, December 2007. doi: 10.1109/TIT.2007.909166.

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

A. D. Scott and A. D. Sokal. The repulsive lattice gas, the independent-set polynomial, and the lovasz local lemma. *Journal of Statistical Physics*, 118:1151–1261, 2005.

Nobuyuki Taga and Shigeru Mase. Error bounds between marginal probabilities and beliefs of loopy belief propagation algorithm. In *MICAI*, pages 186–196, 2006. URL `http://dx.doi.org/10.1007/11925231_18`.

S. C. Tatikonda. Convergence of the sum-product algorithm. In *Proceedings 2003 IEEE Information Theory Workshop*, pages 222–225, April 2003.

S. C. Tatikonda and M. I. Jordan. Loopy belief propogation and Gibbs measures. In *Proc. of the 18th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-02)*, pages 493–500, San Francisco, CA, 2002. Morgan Kaufmann Publishers.

M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, July 2005.

D. Weitz. Counting independent sets up to the tree threshold. In *Proceedings ACM symposium on Theory of Computing*, page 140149. ACM, 2006.

B. Wemmenhove, J. M. Mooij, W. Wiegerinck, M. Leisink, H. J. Kappen, and J. P. Neijt. Inference in the Promedas medical expert system. In *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 2007)*, volume 4594 of *Lecture Notes in Computer Science*, pages 456–460. Springer, 2007. ISBN 978-3-540-73598-4. doi: 10.1007/978-3-540-73599-1_61.