Bounding the Partition Function using Hölder's Inequality

Qiang Liu QLIU1@UCI.EDU
Alexander Ihler IHLER@ICS.UCI.EDU
Department of Computer Science, University of California, Irvine, CA, 92697, USA

Abstract

We describe an algorithm for approximate inference in graphical models based on Hölder's inequality that provides upper and lower bounds on common summation problems such as computing the partition function or probability of evidence in a graphical model. Our algorithm unifies and extends several existing approaches, including variable elimination techniques such as mini-bucket elimination and variational methods such as tree reweighted belief propagation and conditional entropy decomposition. We show that our method inherits benefits from each approach to provide significantly better bounds on sum-product tasks.

1. Introduction

Probabilistic graphical models provide a powerful set of tools for machine learning. A major difficulty in many learning and inference tasks is to calculate the partition function, or normalization constant of the distribution. This task corresponds to computing the probability of evidence in Bayesian networks, and is a critical component of learning a model from data. For trees, the partition function can be calculated efficiently by variable elimination, but for models with cycles this calculation is exponential in the tree width. This makes approximations, particularly upper or lower bounds, of great interest. Examples include approximate elimination methods such as mini-bucket (Dechter & Rish, 2003), and variational approximations such as tree-reweighted belief propagation (TRBP) (Wainwright et al., 2005).

TRBP exploits convexity to provide a bound in terms of a collection of spanning trees. The number of possible spanning trees is extremely large, so direct optimization of the bound is difficult; instead Lagrangian duality is used to derive a loopy fixed-point update, and the tree "weights" are optimized by gradient ascent. Generalized TRBP (Wiegerinck, 2005) extends TRBP to collections of hypertrees, but selecting hypertrees and weights is difficult and typical implementations use only trees and random, unoptimized weights (Mooij, 2010; Schmidt, 2007).

In contrast, mini-bucket elimination is a simple relaxation of variable elimination. It provides either an upper or lower bound and is fast and non-iterative. Computation is controlled with a simple clique size parameter; results can be poor with small cliques, but larger cliques are easily used to improve its bounds.

We propose a generalization of mini-bucket based on Hölder's inequality. We show that the dual form of our bound is equivalent to CED or generalized TRBP, closely connecting variational algorithms with mini-bucket. However, our primal representation inherits many of the benefits of *both* approaches, giving significant advantages over either. In experiments, we show that our method effectively trades off iterative updates with clique size to greatly improve bound quality.

2. Graphical Models

Graphical models capture the factorization structure of a distribution over a collection of variables. Let

$$p(x) = \frac{1}{Z} \prod_{\alpha \in \mathcal{I}} \psi_\alpha(x_\alpha), \tag{1}$$

where α indexes subsets of variables, and Z is a normalizing constant, called the *partition function*. We associate $p(x)$ with a graph $G = (V, E)$, where each variable x_i, $i = 1 \ldots n$ is associated with a node $i \in V$ and $(ij) \in E$ if $\{x_i, x_j\} \subseteq \alpha$ for some α. The set $\mathcal{I}$ is then the set of cliques (fully connected subgraphs) of G. Calculation of Z is a central problem in many learning and inference tasks, but exact evaluation requires summing over an exponential number of terms.

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

The overcomplete exponential family form of (1) is

$$p(\mathbf{x}|\theta) = \exp\big(\theta(x) - \Phi(\theta)\big), \quad \theta(x) = \textstyle\sum_\alpha \theta_\alpha(\mathbf{x}_\alpha),$$

where $\theta_\alpha = \log(\psi_\alpha)$, and $\Phi(\theta) = \log Z(\theta)$ is the log-partition function. $\Phi(\theta)$ is a convex function of θ and its derivatives equal the marginal distributions of $p(x)$.

2.1. Variational Upper Bounds

Tree reweighted BP (Wainwright et al., 2005) builds an upper bound on $\Phi(\theta)$ based on convexity. The idea is to split the parameter vector θ into a sum of vectors $\{\theta^r\}$, $\sum_r \theta^r = \theta$, each of which is a parameter vector of some simpler graphical model T_r (such as a spanning tree). We assign a set of weights $w = [w_1, \cdots, w_R]$ with $\sum_r w_r = 1$, $w_r \geq 0$ and apply Jensen's inequality:

$$\Phi(\theta) = \Phi\Big(\textstyle\sum_r w_r \dfrac{\theta^r}{w_r}\Big) \leq w_r \textstyle\sum_r \Phi\Big(\dfrac{\theta^r}{w_r}\Big) \overset{def}{=} \Psi(\bar{\theta}, w).$$

Finding θ^r and w_r to obtain the tightest upper bound is difficult to solve directly since the number of spanning trees R may be extremely large. Wainwright et al. (2005) provide an elegant alternative by finding a dual formulation of the $\bar{\theta}$-optimal bound for fixed w:

$$\min_{\bar{\theta}} \Psi(\bar{\theta}, w) = \max_{\tau \in \mathbb{L}(G)} \langle \tau, \theta \rangle + \textstyle\sum H(x_i) - \sum \mu_{ij} I_{ij}, \quad (2)$$

where $\mathbb{L}$ is the *locally consistent polytope*, or set of singleton $\tau_i(x_i)$ and pairwise marginals $\tau_{ij}(x_i, x_j)$ that are consistent on their intersections. H_i denotes the entropy of τ_i and I_{ij} the mutual information of τ_{ij}, and μ_{ij} is the so-called edge appearance probability, the sum of weights w_r in trees T_r that contain edge (ij). This dual form can be optimized via a simple "message passing" fixed-point algorithm on G, similar to loopy belief propagation. For w, Wainwright et al. (2005) proposed a conditional gradient method to optimize the μ_{ij} directly, by solving a max-weight spanning tree problem at each iteration. Generalized TRBP (Wainwright et al., 2005; Wiegerinck, 2005) gives a natural extension of TRBP to combinations of hypertrees, similar to generalized BP (Yedidia et al., 2005).

However, a few significant difficulties arise. First, the dual guarantees a valid bound only at message convergence, which may require damping and an unknown number of iterations. Second, it is not obvious how to choose the cliques or "regions" for GTRBP. Finally, weight optimization is significantly more difficult in non-pairwise models; typical implementations of TRBP use randomly generated, unoptimized weights (Mooij, 2010; Schmidt, 2007).

Conditional entropy decomposition (CED) (Globerson & Jaakkola, 2007) extends GTRBP by working purely in the dual, using collections of "elimination orders" (which we call CED-orders, as they consist of both an elimination order and a choice of parent subset for each variable) to define a more general bound. Although potentially tighter, this has similar drawbacks to GTRBP: the dual may not be a bound before convergence, and the desired set of CED-orders must be enumerated and their weights optimized. Here we mainly focus discussion on TRBP, as it is the most common of this general class of variational bounds.

2.2. Bucket Elimination and Mini-Bucket

On the other hand, elimination methods such as bucket or variable elimination (Dechter, 1999; Koller & Friedman, 2009) act by directly summing out the variables in sequence. Bucket elimination assumes a fixed elimination order, and groups factors by placing each ψ in the "bucket" B_i of its earliest argument x_i in the order. We assume without loss of generality that the elimination order is $o = [x_1, \cdots, x_n]$. To illustrate, we use a simple running example with pairwise factors: $p(x) = \frac{1}{Z}\psi_{12}\psi_{13}\psi_{14}\psi_{23}\psi_{24}\psi_{34}$. Its buckets are

B_1: $\{\psi_{12}, \psi_{13}, \psi_{14}\}$ B_2: $\{\psi_{23}, \psi_{24}\}$ B_3: $\{\psi_{34}\}$ B_4: $\{\}$

Inference proceeds by multiplying all factors in B_i and eliminating x_i, resulting in a new factor over all other arguments of factors in B_i; this new factor is then added to the bucket of its "parent" or earliest uneliminated variable, denoted $\mathrm{pa}(i)$. This process is easily seen as applying a distributive property, where each group of factors corresponds to a bucket:

$$Z = \sum_{x_4}\sum_{x_3}\psi_{34}\sum_{x_2}\psi_{23}\psi_{24}\sum_{x_1}\psi_{12}\psi_{13}\psi_{14}.$$

The computational cost of bucket elimination is exponential in the *induced width* of the graph given the elimination order. Mini-bucket (Dechter & Rish, 2003) is an approximation algorithm deriving lower and upper bounds that builds on bucket elimination.

We illustrate mini-bucket with our example. Consider the first bucket B_1, where a sum is performed for each tuple (x_2, x_3, x_4), with total cost $O(d^4)$, where d is the number of possible states of x_i. In general, the computational complexity is $O(d^\omega)$, where ω is the number of variables in each summation (bounded by the induced width of this order). We can instead split the bucket into several smaller "mini-buckets", and eliminate separately; Dechter & Rish (2003) used the inequalities

$$\sum_{x_1^1}\psi_{12}(x_1^1)\psi_{13}(x_1^1)\cdot\min_{x_1^2}\psi_{14}(x_1^2) \;\leq\; \sum_{x_1}\psi_{12}\psi_{13}\psi_{14}$$
$$\leq\; \sum_{x_1^1}\psi_{12}(x_1^1)\psi_{13}(x_1^1)\;\cdot\;\max_{x_1^2}\psi_{14}(x_1^2),$$

where the `sum` and `max` in the bounds are performed for each value of (x_2, x_3) and x_4 separately, having complexity only $O(d^3)$. For more than two mini-buckets, `sum` is used on one and `max` applied to all others.

Mini-bucket uses these bounds to approximate the sum in each bucket, placing the resulting factors in subsequent buckets. We terminate when all buckets are processed. Usually, a parameter *ibound* defines the maximal number of variables allowed in each mini-bucket, controlling the cost-accuracy tradeoff; smaller *ibound*s have less cost, but are typically less accurate.

Mini-bucket is simple, fast, and terminates in a finite (and easy to estimate) number of steps. This enables it to be used with relatively high *ibound* (typically 10–20) to achieve excellent performance. However, it relies primarily on clique size for quality; it cannot be improved iteratively, and few additional heuristics have been proposed (Rollon & Dechter, 2010). Perhaps surprisingly, despite their apparent differences we show that TRBP methods and mini-bucket are closely related, by first generalizing mini-bucket, then showing its connection to TRBP. The resulting algorithm inherits the advantages of both styles, giving considerable power to trade-off performance factors.

3. Hölder's Inequality and WMB

In this section, we introduce Hölder's inequality, and propose a generalization of mini-bucket called *weighted* mini-bucket (WMB). Let $f_i(x)$, $i = 1 \dots n$ be positive functions over a discrete variable x. Let $w = [w_1, w_2, \cdots, w_n]$ be a vector of weights. We define a *weighted summation* (or power sum), given by

$$\sum_x^{w_i} f_i(x) \overset{def}{=} \big(\sum_x f_i(x)^{1/w_i}\big)^{w_i}.$$

Let $w_0 = \sum_i w_i$. If $w_i > 0$ for $i \geq 1$, Hölder's inequality (Hardy et al., 1988) states that

$$\sum_x^{w_0} \prod_i f_i(x) \leq \prod_i \sum_x^{w_i} f_i(x). \qquad (3)$$

If only one weight is positive, e.g., $w_1 > 0$ and $w_i < 0$ for all $i > 1$, we have the reverse Hölder's inequality, where the direction of the bound changes.

The Hölder and reversed Hölder inequalities can be used to provide upper (resp. lower) bounds for the partition function similar to mini-bucket. Following our running example, we assign weights $w = [w_1, w_2]$ satisfying $w_1 + w_2 = 1$ on the copies of variable x_1, and generalize the mini-bucket bound to

$$Z \leq \sum_{x_2,x_3,x_4} \psi_{34}\psi_{23}\psi_{24} \sum_{x_1}^{w_1} \psi_{12}(x_1^1)\psi_{13}(x_1^1) \sum_{x_1}^{w_2} \psi_{14}(x_1^2)$$

Algorithm 1 (Weighted) Mini-bucket Elimination

```
**Input:** The factors of the graphical model $F = \{\psi_\alpha(x_\alpha)\}$, an elimination order $o$, and *ibound*
**Output:** an upper or lower bound on $Z$
for $i \leftarrow 1$ to $n$ do
    $B_i \leftarrow \{\psi_\alpha | \psi_\alpha \in F, i \in \alpha\}$, $F \leftarrow F - B_i$
    Partition $B_i$ into $R_i$ subgroups $\bar{B}_i = \cup_{r=1}^{R_i} B_{i^r}$, such that $|\cup\, var(B_{i^r})| \leq ibound + 1$ for all $r$
    Assign subgroup $r$ weight $w_{i^r}$, where $\sum_r w_{i^r} = 1$
    $F \leftarrow F \cup \{\sum_{x_{i^r}}^{w_{i^r}} \prod_{\psi \in B_{i^r}} \psi\}$
end for
The partition function bound is $\hat{Z} = \prod_{\psi \in F} \psi$.
*Note:* $w_i^r$ are set to 1 or $0^+/0^-$ in standard mini-bucket.
```

for $w_1 > 0$ and $w_2 > 0$; for $w_1 > 0, w_2 < 0$ or $w_1 < 0, w_2 > 0$, the bound reverses to provide a lower bound. Note that $\lim_{w \to 0+} \sum_x^w f(x) = \max_x f(x)$ and $\lim_{w \to 0-} \sum_x^w f(x) = \min_x f(x)$, so the mini-bucket bounds correspond to a limiting case of Hölder's bounds when $w_2 \to 0+$ or $w_2 \to 0-$.

Weighted mini-bucket (WMB) replaces the naïve mini-bucket bound with Hölder's inequality. The same algorithm is used, but `sum/max` are replaced by weighted sum operators, whose weights sum to one on the set of replicates of variables that are split; see Algorithm 1.

Due to limited space, here we focus only on the upper bound component. The lower bound can be analyzed similarly, but its variational form is less well-known and analysis is more complex due to non-convexity issues; see Liu & Ihler (2010).

4. Weighted Log-Partition Function

Although formulated as a sequential elimination algorithm, WMB can be considered "inference" using `weighted sum` operators on a "covering graph" made by splitting nodes of G into disconnected replicates. Several related variable splitting views have been applied to max-product (Johnson, 2008; Choi & Darwiche, 2009; Yarkony et al., 2010). We first give a "primal" view of our bound and establish a few properties, then relate it to TRBP through duality (Section 4.3).

4.1. Building a Covering Graph

First, we interpret the *splitting & elimination* process of mini-bucket as a series of *splitting & triangulation* operations on G, where at each step one node is split into several disconnected replicates and adding edges to triangulate their neighboring, un-eliminated nodes. This process leads to a "covering graph" $\overline{G} = (\bar{V}, \bar{E})$, with $\bar{V} = \{i^r | i \in G, r = 1, \cdots, R_i\}$ being the set of replicates, where i^r is r-th replicate of node i, and R_i is the number of replicates of i. See Fig. 1 for an example. We use "bar" notation to represent the variables and parameters of the new, augmented graph.

The WMB bound can be expressed as a sequence of weighted sums over factors of the covering graph. For notation, we split each x_i in G into R_i replicates $\bar{\mathbf{x}}_i = [x_{i^r}]_{r=1}^{R_i}$ with weights $\bar{\mathbf{w}}_i = [w_{i^r}]_{r=1}^{R_i}$, where $\sum_r w_{i^r} = 1$. Let $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_n]$, $\bar{\mathbf{w}} = [\bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_n]$. The natural extension of elimination order $o = [1 \ldots n]$ on G to $\bar{G}$ is $\bar{o} = [1^1, \cdots, 1^{R_1}; 2^1 \cdots 2^{R_2}; \cdots; n^1 \cdots n^{R_n}]$. For convenience, we also introduce a linear index $\bar{o} = [1, \cdots, k \cdots, \bar{n}]$ to represent $\bar{o}$, so that $i^r \in \bar{V}$ and $k \in \bar{V}$ are two ways to refer to the nodes in $\bar{G}$. Let c_k^0 be the set of neighbors subsequent to node $k \in \bar{V}$ in order $\bar{o}$, and define $c_k = \{k\} \cup c_k^0$.

This transformation is notationally complex, but simple in concept and practice: we simply follow a mini-bucket elimination, keeping track of what elimination operation produced each factor. Set c_k^0 equals the variables in the newly created factor; we again define a "parent" $\overline{\mathrm{pa}}(k) = \min\{c_k^0\}$ to indicate the bucket in which that factor is placed.

We create a parameter vector $\bar{\theta}$ on $\overline{G}$, and require two conditions of the resulting collection of weights and parameters $\bar{\theta}$ (or equivalently factors $\bar{\psi}$) over variable copies $\bar{\mathbf{x}}$. The weights obey sum and sign constraints,

$$\bar{\mathbf{w}} \in \mathcal{D}(w) = \{\bar{\mathbf{w}} | \textstyle\sum_r w_{i^r} = 1, w_{i^r} \geq 0\}.$$

Second, we require the parameters $\bar{\theta}$ be equivalent to the original θ when all replicate copies are equal:

$$\bar{\theta} \in \mathcal{D}(\theta) = \{\bar{\theta} | \bar{\theta}(\bar{\mathbf{x}}) = \theta(\mathbf{x}), \text{when } x_{i^r} \equiv x_i \, \forall i, r\}.$$

4.2. The Weighted Log-Partition Function

In general, the WMB bound has the form

$$\bar{Z}_w(\bar{\theta}, \bar{\mathbf{w}}) = \sum_{\bar{x}_{\bar{n}}}^{\bar{w}_{\bar{n}}} \cdots \sum_{\bar{x}_1}^{\bar{w}_1} \exp(\bar{\theta}(\bar{\mathbf{x}})) \overset{def}{=} \sum_{\bar{\mathbf{x}}}^{\bar{\mathbf{w}}} \exp(\bar{\theta}(\bar{\mathbf{x}})),$$

where the sum follows order $\bar{o}$. (Unlike `sum`, weighted sum operators are not commutative.) We call $\bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}}) = \log \bar{Z}_w(\bar{\theta}, \bar{\mathbf{w}})$ the weighted log-partition function of $\bar{p}(\bar{\mathbf{x}}|\bar{\theta})$. The tightest such bound is

$$\min_{\bar{\theta}, \bar{\mathbf{w}}} \bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}}) \quad \text{s.t.} \quad \bar{\theta} \in \mathcal{D}(\theta), \bar{\mathbf{w}} \in \mathcal{D}(w)$$

It turns out that $\bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}})$ has properties similar to Φ that are useful for solving the optimization; we state these without proof due to limited space. First, as the partition function is defined by $p(x)$, we can conversely define a new distribution $\bar{q}(\bar{\mathbf{x}})$ in terms of the weighted partition function $\bar{Z}_w$. This is defined by a chain rule expansion:

$$\bar{q}(\bar{\mathbf{x}}|\bar{\theta}, \bar{\mathbf{w}}) = \prod_{k=1}^{\bar{n}} \bar{q}(\bar{x}_k | \bar{x}_{k+1:\bar{n}}; \bar{\theta}, \bar{\mathbf{w}}), \tag{4}$$

$$\bar{q}(\bar{x}_k | \bar{x}_{k+1:\bar{n}}; \bar{\theta}, \bar{\mathbf{w}}) \propto \Big[\sum_{\bar{x}_{1:k-1}}^{\bar{w}_{1:k-1}} \exp(\bar{\theta}(\bar{\mathbf{x}})) \Big]^{1/\bar{w}_k}$$

We then have the following result:

Theorem 4.1. *The derivatives of $\bar{\Phi}_w$ correspond to the moments and conditional entropies of $\bar{q}(\cdot)$, i.e.,*

$$\frac{\partial}{\partial \bar{\theta}_\alpha} \bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}}) = \bar{q}(\bar{\mathbf{x}}_\alpha | \bar{\theta}, \bar{\mathbf{w}}), \tag{5}$$

$$\frac{\partial}{\partial \bar{\mathbf{w}}_k} \bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}}) = H_w(\bar{x}_k | \bar{x}_{k+1:\bar{n}} \,;\, \bar{q}), \tag{6}$$

where $\bar{q}(\bar{\mathbf{x}}_\alpha | \bar{\theta}, \bar{\mathbf{w}})$ is the marginal of $\bar{q}(\bar{\mathbf{x}} | \bar{\theta}, \bar{\mathbf{w}})$ and $H_w(\cdot|\cdot; \bar{q})$ is the conditional entropy of $\bar{q}$:

$$H_w(\bar{x}_k | \bar{x}_{k+1:\bar{n}} \,;\, \bar{q}) = -\sum_{\bar{\mathbf{x}}} \bar{q}(\bar{x}_{k:\bar{n}}) \log \bar{q}(\bar{x}_k | \bar{x}_{k+1:\bar{n}}).$$

Further, $\bar{\Phi}_w$ is jointly convex w.r.t. $[\theta, \bar{\mathbf{w}}]$ when $\bar{\mathbf{w}} > 0$.

Algorithm 2 gives a message-passing computation for these quantities. The forward pass exactly equals the WMB of Algorithm 1 and $\bar{\Phi}_w$ its bound; the backward pass computes the marginals $\bar{q}(\cdot)$. This procedure can be viewed as message-passing on the junction tree of cliques $\mathcal{C} = \{c_k\}$, with hyperedges connecting c_k and $c_{\overline{\mathrm{pa}}(k)}$. Since the induced width of $\overline{G}$ is bounded by *ibound*, the total complexity is $O(d^{ibound})$.

Algorithm 2 Inference equations for $\bar{\Phi}_w$

For node $k \in \bar{V}$, parent $l = \overline{\mathrm{pa}}(k)$ and neighbors $\sim k$:

Forward: (*weighted mini-bucket elimination*)

$$m_{k \to l}(\bar{\mathbf{x}}_{c_l}) = \sum_{\bar{x}_k}^{\bar{w}_k} \bar{\psi}_{c_k} \prod_{j: k \in \overline{\mathrm{pa}}(j)} m_{j \to k} \tag{7}$$

Backward: (*to compute marginals, gradients*)

$$m_{l \to k} = \Big[\sum_{\bar{\mathbf{x}}_{c_l / c_k}} (\bar{\psi}_{c_l} m_{\sim l})^{1/\bar{w}_l} m_{k \to l}^{-1/\bar{w}_k} \Big]^{\bar{w}_k} \tag{8}$$

Note that backward messages depend on forward ones, but not vice versa. Here $\bar{\psi}_{c_k} = \exp(\bar{\theta}_{c_k})$.

Marginals: $\bar{q}(x_{c_k}) \propto (\bar{\psi}_{c_k} m_{\sim k})^{1/w_k}$ (9)

Bound:
$$\bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}}) = \log \prod_{k: \overline{\mathrm{pa}}(k) = \emptyset} \sum_{\bar{x}_k}^{\bar{w}_k} \bar{\psi}_{c_k} m_{\sim k} \tag{10}$$

4.3. Duality and Connection to TRBP

We now describe the dual representation of $\bar{\Phi}_w(\bar{\theta}, \bar{w})$, again parallel to the regular log-partition function. Note that in this section we consider $\bar{\Phi}_w$ as a function of $\bar{\theta}$, for fixed positive weights $\bar{\mathbf{w}} > 0$.

Theorem 4.2. (1). *$\bar{\Phi}_w(\bar{\theta}, \bar{\mathbf{w}})$ is a convex function of $\bar{\theta}$ for fixed $\bar{\mathbf{w}} > 0$, and has dual representation,*

$$\bar{\Phi}_w(\bar{\theta}) = \max_{\bar{\tau} \in \mathbb{M}(\overline{G})} \langle \bar{\tau}, \bar{\theta} \rangle + \sum_{k=1}^{\bar{n}} \bar{w}_k H_w(\bar{x}_k | \bar{x}_{c_k^0}\,; \bar{\tau}), \quad (11)$$

where $\mathbb{M}(\overline{G})$ is the marginal polytope on $\overline{G}$, and $H_w(\bar{x}_k | \bar{x}_{c_k^0}\,; \bar{\tau})$ is the conditional entropy under $\bar{\tau}$. The maximum is obtained iff $\bar{\tau}(\bar{\mathbf{x}}) = \bar{q}(\bar{\mathbf{x}} | \bar{\theta}, \bar{\mathbf{w}})$.

(2). *Let $\Gamma : \bar{V} \to V$ map replicates to their original nodes, i.e., $\Gamma(i^r) = i$. For fixed $\bar{\mathbf{w}} > 0$, we have*

$$\min_{\theta \in \mathcal{D}(\theta)} \bar{\Phi}_w(\bar{\theta}) = \max_{\tau \in \mathbb{L}(G)} \langle \tau, \theta \rangle + \sum_{k=1}^{\bar{n}} \bar{w}_k H(x_{\Gamma(k)} | \bar{x}_{\Gamma(c_k^0)}\,; \tau)$$

where $\mathbb{L}(G)$ is the local polytope for beliefs τ defined over sets of variables $\Gamma(c_k)$ in the original graph G.

Theorem 4.2(1) gives the dual form of the weighted log-partition function. When $\bar{w}_k = 1$ for all k, this reduces to the standard dual of the log-partition function (Wainwright & Jordan, 2008). Note that since $\overline{G}$ is triangulated, $\mathbb{M}(\overline{G})$ is also the collection of $\bar{\tau}$ that are consistent on the intersections of cliques $\{c_k\}_{k=1}^{\bar{n}}$.

Theorem 4.2(2) is a corollary of Theorem 4.2(1), which gives a dual approach for finding the $\bar{\theta}$-optimal bound. It has the form of a weighted sum of conditional entropies, and it is not difficult to show that it is equivalent to some set of CED-orders consistent with elimination order o, and is thus a form of CED; if these conditional entropies are consistent with some hypertree order (always true for *ibound* = 1), it will also be a form of generalized TRBP, and suggests that one way to optimize $\bar{\theta}$ is via TRBP message passing.

Interestingly, the elementary building blocks of CED are single CED-orders, for which $\bar{w}_k = 1$ or 0^+, which can be seen to correspond to a dual form of mini-bucket with `max`. The WMBE bound for a given covering tree corresponds to some collection of CED-orders or GTRBP hypertrees (we return to this point shortly). Also, any CED-order can be expressed as some covering tree. As CED uses collections of CED-orders, we could elect to use a TRBP-like "weighted collection" of more than one covering tree to achieve as tight a bound as CED. However, here we focus on the expressive power of a single covering tree, which enables a simple and efficient *primal* bound and the use of existing heuristics for clique choice.

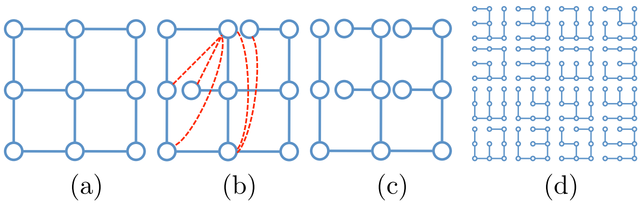


Figure 1. (a) A 3×3 grid. (b) A covering graph with *ibound* 2 (column-first order). (c) A covering tree. (d) The set of spanning tree that is equivalent to the covering tree in (c).

4.4. Primal vs. Dual, Covering vs. Spanning

In addition to closely connecting these algorithms, our technique has significant practical advantages. Although its dual is equivalent to that GTRBP or CED, WMB gives a concise and novel primal bound. The primal form of TRBP is intractable unless the number of spanning trees is small. Even CED's dual form relies on a small number of CED-orders to be tractable, and the primal form derived in Globerson & Jaakkola (2007) (a geometric program) is computationally infeasible. In contrast, WMBE uses only a few parameters to capture the equivalent of a large collection of spanning trees or orders. This set can be made precise, but we illustrate it with a simple example in Fig. 1. For a 3×3 grid, the covering tree in panel (c) gives a bound equivalent to that of the collection of 16 spanning trees in panel (d). In general the covering tree can have many fewer degrees of freedom than its corresponding collection of trees or orders (here, 4α versus 15α).

Not only is this more efficient, it also improves the pre-convergence bound. The "extra" degrees of freedom can only loosen the bound; their removal corresponds to enforcing they take on optimal values. These advantages improve our ability to represent and optimize a primal form of the bound with many spanning trees. Advantages over the dual form include easily optimizing $\bar{\mathbf{w}}$ (in TRBP, the gradient is technically valid only at convergence), the ability to flexibly balance optimizing $\bar{\theta}$, $\bar{\mathbf{w}}$, or increasing the *ibound*, and providing a valid bound at any point (any *any-time* property).

5. Tightening the Primal Bound

Our concise WMB bound lends itself to efficient primal algorithms for optimizing $\bar{\theta}$ and $\bar{\mathbf{w}}$. Generally speaking the optimization is a convex program with simple constraint conditions; the time complexity of the objective function, gradient or Hessian is $O(d^{ibound})$, under user control. We could thus use black-box convex optimization routines to optimize the bound. However, since $\bar{\Phi}_w$ is calculated using message-passing, it seems most effective to update $\bar{\theta}$ and $\bar{\mathbf{w}}$ while computing messages.

Algorithm 3 Tightening the bound $\bar{\Phi}_w$

```
  Initialize $\bar{\theta}$, $\bar{\mathbf{w}}$ and messages $m$
  repeat
    for $i \leftarrow 1$ to $n$ do
      pre-update backward messages (optional):
      update $m_{i^r \to \overline{\mathrm{pa}}(i^r)}$ and then $m_{\overline{\mathrm{pa}}(i^r) \to i^r}$ by (7)-(8) for all replicates of $i$.
      Reallocate / reweight:
      update $\theta$ and/or $\bar{\mathbf{w}}$ by fixed point (12), log-gradient (13) or other methods.
      Update forward messages:
      update $m_{i \to \overline{\mathrm{pa}}(i^r)}$ by (7) for all replicates of $i$.
    end for
    Calculate the bound $\bar{\Phi}_w$ by (10).
    for $i \leftarrow n$ to 1 do
      Reallocate / reweight:
      Update $\theta$ and/or $\bar{\mathbf{w}}$ by fixed point (12), log-gradient (13) or other methods.
      Update backward messages:
      update $m_{\overline{\mathrm{pa}}(i^r) \to i}$ by (8) for all replicates of $i$.
    end for
  until stopping criterion satisfied.
```

A surprisingly powerful version of this is explored in our experiments (Section 6), by non-iteratively optimizing "on the fly" during a single forward pass.

5.1. Optimizing $\bar{\theta}$

Consider a node $i \in V$ and its replicates $\{i^r\}$. For a given initial $\bar{\theta}$, we have freedom to reallocate factors between the replicates, e.g., we can update $\bar{\theta}$ to $\bar{\theta} + \sum_r \vartheta_{i^r}$, where ϑ_{i^r} are singleton factors satisfying $\sum_r \vartheta_{i^r}(x_i) = 0$. The optimal ϑ_{i^r} satisfy a moment matching condition $b_{i^r}(x_i) \equiv b_i(x_i)$, i.e., all replicates share the same marginal; this parallels the matching condition in Wainwright et al. (2005). To select ϑ_{i^r}, we can use the fixed-point update:

Updating $\bar{\theta} = \log \bar{\psi}$:

$$b_{i^r} = \sum_{\bar{\mathbf{x}} \setminus \{x_{i^r}\}} (\bar{\psi}_{c_{i^r}} m_{\sim i^r})^{1/w_{i^r}} \ , \quad b_i = \Big(\prod_r {b_{i^r}}^{w_{i^r}} \Big)^{\frac{1}{\sum_r w_{i^r}}}$$

$$\bar{\psi}_{c_{i^r}} \leftarrow \bar{\psi}_{c_{i^r}} \big(\frac{b_i}{b_{i^r}}\big)^{w_{i^r}} \qquad (12)$$

where b_{i^r} is the marginal of $b_{c_{i^r}}$ in (9), and b_i is the weighted geometric mean of the b_{i^r}. It is easy to show that updates keep $\bar{\theta}$ consistent with θ, and that the fixed point satisfies the marginal matching condition. It is also easily generalized to larger sets of replicates.

Update (12) can be inserted into forward-backward calculations in different ways; Algorithm 3 describes two variants. Since backward messages $m_{\overline{\mathrm{pa}}(i^r) \to i^r}$ depend on the forward messages, updating the forward messages invalidates the backward messages. The optional step of Algorithm 3 "pre-updates" the backward messages before updating $\bar{\theta}$.

Many other update choices are equally valid. For example, a projected gradient step is particularly simple:

$$\theta_{i^r} \leftarrow \theta_{i^r} + \mu(b_{i^r} - b_i),$$

where b_i is the arithmetic mean of b_{i^r}; and Newton-like methods could also improve speed and convergence.

5.2. Optimizing $\bar{\mathbf{w}}$

Unlike the TRBP dual, in which the weight gradient is only correct after $\bar{\theta}$ has been optimized, our algorithm can update $\bar{\mathbf{w}}$ independent of $\bar{\theta}$. In our experiments we found weight updates could be surprisingly important.

By the KKT conditions, the stationary condition is

$$w_{i^r}(H_{i^r|\prec} - H_{i|\prec}) = 0, \forall i^r \in \bar{V},$$

where $H_{i^r|\prec} = H_w(x_{i^r}|x_{c^0_{i^r}})$, and $H_{i|\prec} = \sum_r w_{i^r} H_{i^r|\prec}$ is the average conditional entropy over the replicates. Thus at the stationary point, either the weight is zero or the replicate conditional entropy equals the average.

The constraint set $\mathcal{D}(w_i)$ is simple, and can be transformed to an unconstrained problem by taking $w_{i^r} = \exp(u_{i^r}) / \sum_r \exp(u_{i^r})$, for $u_{i^r} \in \mathbb{R}$. Taking the gradient w.r.t. u_{i^r} and substituting gives log-gradient step:

Updating weights:

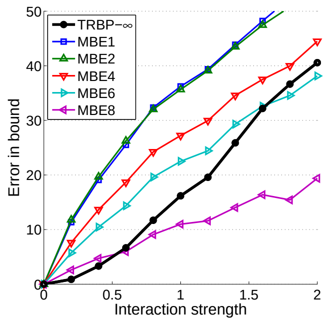
$$\begin{aligned} w_{i^r} &\leftarrow w_{i^r} \exp(-\epsilon \, w_{i^r}(H_{i^r|\prec} - H_{i|\prec})) \\ w_{i^r} &\leftarrow w_{i^r} / \textstyle\sum_r w_{i^r}. \end{aligned} \qquad (13)$$

where ϵ is a step size. This update ensures $\bar{\mathbf{w}}$ in $\mathcal{D}(w)$ and its fixed point satisfies the KKT condition.

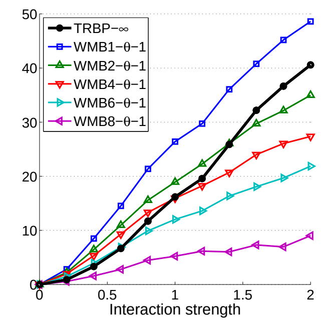
As with $\bar{\theta}$, other optimizations of $\bar{\mathbf{w}}$ are equally valid. The conditional gradient (analogous to the update in Wainwright et al. (2005)) simply finds the replicate with lowest conditional entropy. In our experiments, we use the log-gradient update; in practice it appeared to converge better than conditional gradient.

5.3. Choosing the Covering Structure

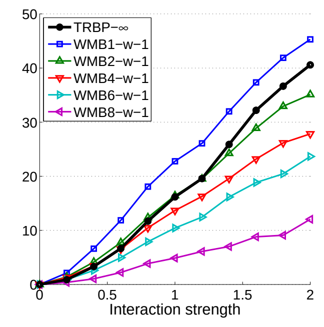
The tightness of the bound also depends on the structure of the covering graph, which stems from the bucket partitioning strategy and elimination order. These elements are harder to optimize, although some heuristics exist (Rollon & Dechter, 2010). For our WMB, we use the simplest scope-based heuristics, which greedily minimize the number of splits required (Dechter & Rish, 2003): each factor is added to an



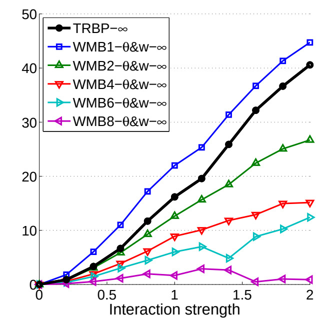
(a) Mini-bucket



(b) θ optimized, one pass



(c) w optimized, one pass



(d) Both fully optimized

Figure 2. Results on 10×10 Ising model with mixed interactions. (a) Mini-bucket is often worse than TRBP except for high *ibound*, but (b),(c) weighted mini-bucket is often better even with only a forward pass. (d) Fully optimized WMB is slightly looser than TRBP for *ibound* $= 1$, but quickly surpasses TRBP as *ibound* increases. See also Fig. 3.

existing mini-bucket if possible, and otherwise placed in a new mini-bucket. We also add a *refill* step after partitioning, which greedily tries to split factors across multiple mini-buckets (i.e., increase the buckets' clique) if doing so does not violate the *ibound*.

6. Experiments

We tested WMB on synthetic networks (Ising models) and on a number of real-world linkage analysis models from the UAI'08 competition to show its effectiveness compared to standard TRBP and mini-bucket.

Ising models. We generated random 10×10 Ising models (binary pairwise grids),

$$p(x) = \exp\Big[\sum_{i \in V} \theta_i x_i + \sum_{(i,j) \in E} \theta_{ij} x_i x_j - \Phi(\theta)\Big],$$

where $x_i \in \{-1, 1\}$ and $\theta_i \sim \mathcal{N}(0, 0.1)$, $\theta_{ij} \sim \mathcal{N}(0, \sigma)$, with $\sigma \in [0, 2]$. (This gives "mixed" interactions; relative performance on purely-attractive models was similar and is omitted for space).

For WMB, we use the column-first order and scope-based heuristic, allocate $\bar{\theta}$ uniformly across replicates, and initialize weights and messages uniformly.

Fig. 2 shows several curves comparing bound quality as σ is varied in $[0, 2]$. In each plot, TRBP-∞ represents TRBP when weights are full optimized using conditional gradient (the optimal bound with tree-width 1). The first panel shows the bound found by naïve mini-bucket (MBE) for various *ibound*s. We then show our proposed WMB bound when either $\bar{\theta}$ or $\bar{\mathbf{w}}$ is modified during a *single* forward pass (so that the estimate has the same computational complexity as mini-bucket), and when the bound is fully optimized. Fig. 3 shows a calibrated timing comparison on a single example (with $\sigma = 1$), in which one unit of time equals a full iteration of TRBP or a single forward-backward pass of WMB. We use a gradient step on μ_{ij} at each iteration of TRBP (step size .1) to find the optimal bound.

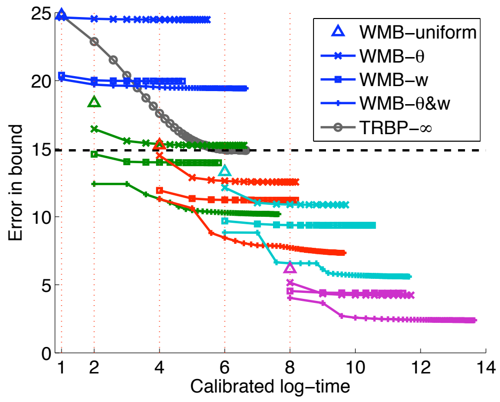
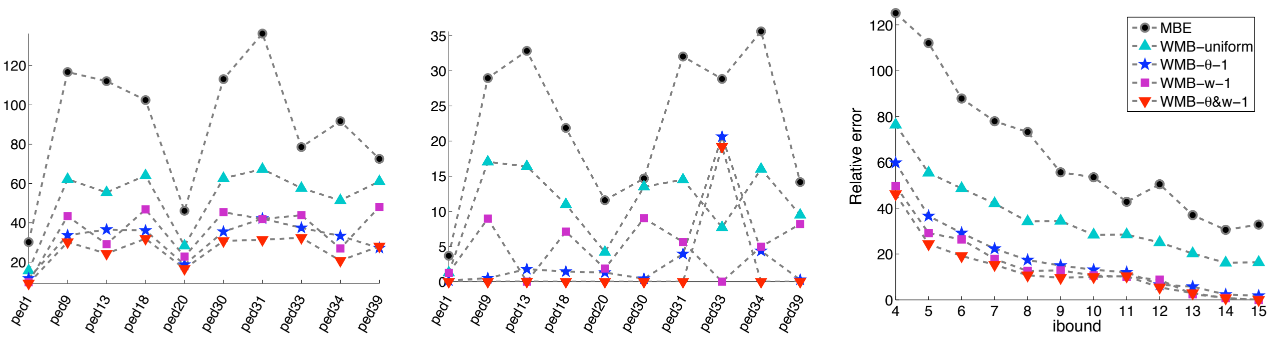


Figure 3. Calibrated timing comparison on a single grid. Colors indicate increasing *ibound* (see Fig. 2). One unit of time is a full iteration of TRBP plus a weight step, or a single forward-backward pass of WMB; trajectories indicate the current tightest bounds. WMB with w-steps out-performs θ-steps in this instance; optimizing both θ&w is usually best. The largest gain is from increasing *ibound*, e.g., `WMB2-`θ&w is better than `TRBP-`∞ in its first iteration.

The most dominant factor in bound quality was *ibound* – for example, `WMB`i-w-1 (indicating one iteration of WMB with weight updates and *ibound* $= i$) is already as good as `TRBP-`∞, the fully-converged TRBP bound, by $i = 2$. Perhaps surprisingly, for Ising models we found that optimizing $\bar{\mathbf{w}}$ consistently gave tighter bounds than optimizing $\bar{\theta}$. The importance of weights is intuitive, since changing their sign (to negative) can actually give a lower bound. However, most implementations of TRBP (Mooij, 2010; Schmidt, 2007) do not include weight optimization even for pairwise models, and it is even more difficult for larger cliques.



(a) *ibound* = 5 (b) *ibound* = 15 (c) `pedigree13`

Figure 4. Linkage analysis networks. (a)-(b) We show the error relative to the tightest bound found on 10 problems at two *ibound*s, for mini-bucket and four single-pass WMB estimates: uniform (no updates), w or θ only, and both. All four consistently outperform standard mini-bucket. (c) A typical instance, showing quality as *ibound* is increased.

Additionally, the most benefit occurred within the first 1-2 iterations of WMB, suggesting that running message-passing to convergence may be wasteful. It appears better to extract a primal WMB bound early, then increase *ibound*. This shows the advantage of the primal bound's *any-time* property: the algorithm can stop prior to convergence and return a valid bound.

Linkage analysis. We also compared our algorithm to standard mini-bucket on models for pedigree linkage analysis from the UAI'08 approximate inference challenge. The models have ∼300-1000 nodes, with induced width of ∼20-30. We compare MBE to WMB with only one forward elimination, giving both methods equal time complexity and varying *ibound* $\in [4 \ldots 15]$. The implementation of MBE follows Rollon & Dechter (2010), which used advanced heuristics for bucket partitioning; our WMB continued to use naïve scope-based heuristics (see section 5.3).

Fig. 4 shows that all versions of WMB significantly outperform mini-bucket. Even `WMB-uniform`, which performs no updates to $\bar{\theta}$ or $\bar{\mathbf{w}}$ (uniform allocation) outperforms MBE. Unlike the Ising models, here we find `WMB-`w is not consistently better than `WMB-`θ.

7. Conclusion

We presented an algorithm that unifies and extends mini-bucket and TRBP. Our algorithm inherits significant benefits from both views: it is able to produce a bound at any point (often a single iteration is sufficient); it compactly represents and can optimize over a large class of TRBP bounds; and it can easily improve its bounds via iterations or increased clique size. Future work includes studying the negative weight lower bound, and exploring better heuristics for selecting elimination orders and constructing covering graphs.

Acknowledgements

Work supported in part by NSF grant IIS-1065618.

References

Choi, A. and Darwiche, A. Relax then compensate: Max-product belief propagation and more. In *NIPS*, 2009.

Dechter, R. and Rish, I. Mini-buckets: A general scheme for bounded inference. *J. ACM*, 50(2):107–153, 2003.

Dechter, Rina. Bucket elimination: a unifying framework for reasoning. *Artif. Intell.*, 113:41–85, September 1999.

Globerson, A. and Jaakkola, T. Approximate inference using conditional entropy decompositions. In *AISTATS*, 2007.

Hardy, G. H., Littlewood, J. E., and Pólya, G. *Inequalities*. Cambridge University Press, 1988.

Johnson, J. *Convex relaxation methods for graphical models: Lagrangian and maximum entropy approaches*. PhD thesis, MIT, August, 2008.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Liu, Q. and Ihler, A. Negative tree reweighted belief propagation. In *UAI*, 2010.

Mooij, J. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, August 2010.

Rollon, E. and Dechter, R. New mini-bucket partitioning heuristics for bounding the probability of evidence. In *AAAI*, 2010.

Schmidt, M. UGM matlab toolbox, 2007. URL `http://people.cs.ubc.ca/~schmidtm/Software/UGM.html`.

Wainwright, M. and Jordan, M. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

Wainwright, M., Jaakkola, T., and Willsky, A. A new class of upper bounds on the log partition function. *IEEE Trans. Info. Theory*, 51(7):2313–2335, July 2005.

Wiegerinck, W. Approximations with reweighted generalized belief propagation. In *AISTATS*, 2005.

Yarkony, J., Fowlkes, C., and Ihler, A. Covering trees and lower bounds on quadratic assignment. In *CVPR*, 2010.

Yedidia, J.S., Freeman, W.T., and Weiss, Y. and. Constructing free-energy approximations and generalized BP algorithms. *IEEE Trans. Inf. Theory*, 51, July 2005.