# Decomposition Bounds for Marginal MAP

**Wei Ping**[*]     **Qiang Liu**[†]     **Alexander Ihler**[*]
[*]Computer Science, UC Irvine     [†]Computer Science, Dartmouth College
{wping,ihler}@ics.uci.edu   qliu@cs.dartmouth.edu

## Abstract

Marginal MAP inference involves making MAP predictions in systems defined with latent variables or missing information. It is significantly more difficult than pure marginalization and MAP tasks, for which a large class of efficient and convergent variational algorithms, such as dual decomposition, exist. In this work, we generalize dual decomposition to a generic *power sum* inference task, which includes marginal MAP, along with pure marginalization and MAP, as special cases. Our method is based on a block coordinate descent algorithm on a new convex decomposition bound, that is guaranteed to converge monotonically, and can be parallelized efficiently. We demonstrate our approach on marginal MAP queries defined on real-world problems from the UAI approximate inference challenge, showing that our framework is faster and more reliable than previous methods.

## 1   Introduction

Probabilistic graphical models such as Bayesian networks and Markov random fields provide a useful framework and powerful tools for machine learning. Given a graphical model, *inference* refers to answering probabilistic queries about the model. There are three common types of inference tasks. The first are max-inference or maximum a *posteriori* (MAP) tasks, which aim to find the most probable state of the joint probability; exact and approximate MAP inference is widely used in structured prediction [26]. Sum-inference tasks include calculating marginal probabilities and the normalization constant of the distribution, and play a central role in many learning tasks (e.g., maximum likelihood). Finally, marginal MAP tasks are "mixed" inference problems, which generalize the first two types by marginalizing a subset of variables (e.g., hidden variables) before optimizing over the remainder.[1] These tasks arise in latent variable models [e.g., 29, 25] and many decision-making problems [e.g., 13]. All three inference types are generally intractable; as a result, approximate inference, particularly convex relaxations or upper bounding methods, are of great interest.

Decomposition methods provide a useful and computationally efficient class of bounds on inference problems. For example, dual decomposition methods for MAP [e.g., 31] give a class of easy-to-evaluate upper bounds which can be directly optimized using coordinate descent [36, 6], subgradient updates [14], or other methods [e.g., 22]. It is easy to ensure both convergence, and that the objective is monotonically decreasing (so that more computation always provides a better bound). The resulting bounds can be used either as stand-alone approximation methods [6, 14], or as a component of search [11]. In summation problems, a notable decomposition bound is tree-reweighted BP (TRW), which bounds the partition function with a combination of trees [e.g., 34, 21, 12, 3]. These bounds are useful in joint inference and learning (or "inferning") frameworks, allowing learning with approximate inference to be framed as a joint optimization over the model parameters and decomposition bound, often leading to more efficient learning [e.g., 23]. However, far fewer methods have been developed for marginal MAP problems.

---

[1]In some literature [e.g., 28], marginal MAP is simply called MAP, and the joint MAP task is called MPE.

In this work, we deveop a decomposition bound that has a number of desirable properties: (1) *Generality*: our bound is sufficiently general to be applied easily to marginal MAP. (2) *Any-time*: it yields a bound at any point during the optimization (not just at convergence), so it can be used in an anytime way. (3) *Monotonic and convergent*: more computational effort gives strictly tighter bounds; note that (2) and (3) are particularly important for high-width approximations, which are expensive to represent and update. (4) Allows *optimization over all parameters*, including the "weights", or fractional counting numbers, of the approximation; these parameters often have a significant effect on the tightness of the resulting bound. (5) *Compact representation*: within a given class of bounds, using fewer parameters to express the bound reduces memory and typically speeds up optimization.

We organize the rest of the paper as follows. Section 2 gives some background and notation, followed by connections to related work in Section 3. We derive our decomposed bound in Section 4, and present a (block) coordinate descent algorithm for monotonically tightening it in Section 5. We report experimental results in Section 6 and conclude the paper in Section 7.

## 2   Background

Here, we review some background on graphical models and inference tasks. A Markov random field (MRF) on discrete random variables $x = [x_1, \ldots, x_n] \in \mathcal{X}^n$ is a probability distribution,

$$p(x; \theta) = \exp \Big[ \sum_{\alpha \in \mathcal{F}} \theta_\alpha(x_\alpha) - \Phi(\theta) \Big]; \quad \Phi(\theta) = \log \sum_{x \in \mathcal{X}^n} \exp \Big[ \sum_{\alpha \in \mathcal{F}} \theta_\alpha(x_\alpha) \Big], \quad (1)$$

where $\mathcal{F}$ is a set of subsets of the variables, each associated with a factor $\theta_\alpha$, and $\Phi(\theta)$ is the log partition function. We associate an undirected graph $G = (V, E)$ with $p(x)$ by mapping each $x_i$ to a node $i \in V$, and adding an edge $ij \in E$ iff there exists $\alpha \in \mathcal{F}$ such that $\{i, j\} \subseteq \alpha$. We say node $i$ and $j$ are neighbors if $ij \in E$. Then, $\mathcal{F}$ is the subset of cliques (fully connected subgraphs) of $G$.

The use and evaluation of a given MRF often involves different types of inference tasks. *Marginalization*, or *sum-inference* tasks perform a sum over the configurations to calculate the log partition function $\Phi$ in (1), marginal probabilities, or the probability of some observed evidence. On the other hand, the maximum *a posteriori* (MAP), or *max-inference* tasks perform joint maximization to find configurations with the highest probability, that is, $\Phi_0(\theta) = \max_x \sum_{\alpha \in \mathcal{F}} \theta_\alpha(x_\alpha)$.

A generalization of max- and sum- inference is *marginal MAP*, or *mixed-inference*, in which we are interested in first marginalizing a subset $A$ of variables (e.g., hidden variables), and then maximizing the remaining variables $B$ (whose values are of direct interest), that is,

$$\Phi_{AB}(\theta) = \max_{x_B} Q(x_B) = \max_{x_B} \log \sum_{x_A} \exp \Big[ \sum_{\alpha \in \mathcal{F}} \theta_\alpha(x_\alpha) \Big], \quad (2)$$

where $A \cup B = V$ (all the variables) and $A \cap B = \emptyset$. Obviously, both sum- and max- inference are special cases of marginal MAP when $A = V$ and $B = V$, respectively.

It will be useful to define an even more general inference task, based on a power sum operator:

$$\sum_{x_i}^{\tau_i} f(x_i) = \Big[ \sum_{x_i} f(x_i)^{1/\tau_i} \Big]^{\tau_i},$$

where $f(x_i)$ is any non-negative function and $\tau_i$ is a *temperature* or *weight* parameter. The power sum reduces to a standard sum when $\tau_i = 1$, and approaches $\max_x f(x)$ when $\tau_i \to 0^+$, so that we define the power sum with $\tau_i = 0$ to equal the max operator.

The power sum is helpful for unifying max- and sum- inference [e.g., 35], as well as marginal MAP [15]. Specifically, we can apply power sums with different weights $\tau_i$ to each variable $x_i$ along a predefined elimination order (e.g., $[x_1, \ldots, x_n]$), to define the *weighted log partition function*:

$$\Phi_{\boldsymbol{\tau}}(\theta) = \log \sum_x^{\boldsymbol{\tau}} \exp(\theta(x)) = \log \sum_{x_n}^{\tau_n} \ldots \sum_{x_1}^{\tau_1} \exp(\theta(x)), \quad (3)$$

where we note that the value of (3) depends on the elimination order unless all the weights are equal. Obviously, (3) includes marginal MAP (2) as a special case by setting weights $\tau_A = 1$ and $\tau_B = 0$. This representation provides a useful tool for understanding and deriving new algorithms for general inference tasks, especially marginal MAP, for which relatively few efficient algorithms exist.

2

## 3 Related Work

Variational upper bounds on MAP and the partition function, along with algorithms for providing fast, convergent optimization, have been widely studied in the last decade. In MAP, dual decomposition and linear programming methods have become a dominating approach, with numerous optimization techniques [36, 6, 32, 14, 37, 30, 22], and methods to tighten the approximations [33, 14].

For summation problems, most upper bounds are derived from the tree-reweighted (TRW) family of convex bounds [34], or more generally conditional entropy decompositions [5]. TRW bounds can be framed as optimizing over a convex combination of tree-structured models, or in a dual representation as a message-passing, TRW belief propagation algorithm. This illustrates a basic tension in the resulting bounds: in its primal form [2] (combination of trees), TRW is inefficient: it maintains a weight and $O(|V|)$ parameters for each tree, and a large number of trees may be required to obtain a tight bound; this uses memory and makes optimization slow. On the other hand, the dual, or free energy, form uses only $O(|E|)$ parameters (the TRW messages) to optimize over the set of all possible spanning trees – but, the resulting optimization is only guaranteed to be a bound at convergence, [3] making it difficult to use in an anytime fashion. Similarly, the gradient of the weights is only correct at convergence, making it difficult to optimize over these parameters; most implementations [e.g., 24] simply adopt fixed weights.

Thus, most algorithms do not satisfy all the desirable properties listed in the introduction. For example, many works have developed convergent message-passing algorithms for convex free energies [e.g., 9, 10]. However, by optimizing the dual they do not provide a bound until convergence, and the representation and constraints on the counting numbers do not facilitate optimizing the bound over these parameters. To optimize counting numbers, [8] adopt a more restrictive free energy form requiring positive counting numbers on the entropies; but this cannot represent marginal MAP, whose free energy involves conditional entropies (equivalent to the difference between two entropy terms).

On the other hand, working in the primal domain ensures a bound, but usually at the cost of enumerating a large number of trees. [12] heuristically select a small number of trees to avoid being too inefficient, while [21] focus on trying to speed up the updates on a given collection of trees. Another primal bound is weighted mini-bucket (WMB, [16]), which can represent a large collection of trees compactly and is easily applied to marginal MAP using the weighted log partition function viewpoint [15, 18]; however, existing optimization algorithms for WMB are non-monotonic, and often fail to converge, especially on marginal MAP tasks.

While our focus is on variational bounds [16, 17], there are many non-variational approaches for marginal MAP as well. [27, 38] provide upper bounds on marginal MAP by reordering the order in which variables are eliminated, and using exact inference in the reordered join-tree; however, this is exponential in the size of the (unconstrained) treewidth, and can easily become intractable. [20] give an approximation closely related to mini-bucket [2] to bound the marginal MAP; however, unlike (weighted) mini-bucket, these bounds cannot be improved iteratively. The same is true for the algorithm of [19], which also has a strong dependence on treewidth. Other examples of marginal MAP algorithms include local search [e.g., 28] and Markov chain Monte Carlo methods [e.g., 4, 39].

## 4 Fully Decomposed Upper Bound

In this section, we develop a new general form of upper bound and provide an efficient, monotonically convergent optimization algorithm. Our new bound is based on fully decomposing the graph into disconnected cliques, allowing very efficient local computation, but can still be as tight as WMB or the TRW bound with a large collection of spanning trees once the weights and shifting variables are chosen or optimized properly. Our bound reduces to dual decomposition for MAP inference, but is applicable to more general mixed-inference settings.

Our main result is based on the following generalization of the classical Hölder's inequality [7]:

---

[2]Despite the term "dual decomposition" used in MAP tasks, in this work we refer to decomposition bounds as "primal" bounds, since they can be viewed as directly bounding the result of variable elimination. This is in contrast to, for example, the linear programming relaxation of MAP, which bounds the result only after optimization.

[3]See an example on Ising model in Supplement A.

**Theorem 4.1.** *For a given graphical model $p(x; \theta)$ in (1) with cliques $\mathcal{F} = \{\alpha\}$ and a set of non-negative weights $\boldsymbol{\tau} = \{\tau_i \geq 0, i \in V\}$, we define a set of "split weights" $\mathbf{w}^\alpha = \{w_i^\alpha \geq 0, i \in \alpha\}$ on each variable-clique pair $(i, \alpha)$, that satisfies $\sum_{\alpha | \alpha \ni i} w_i^\alpha = \tau_i$. Then we have*

$$\sum_x^{\boldsymbol{\tau}} \prod_{\alpha \in \mathcal{F}} \exp\left[\theta_\alpha(x_\alpha)\right] \leq \prod_{\alpha \in \mathcal{F}} \sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp\left[\theta_\alpha(x_\alpha)\right], \tag{4}$$

*where the left-hand side is the powered-sum along order $[x_1, \ldots, x_n]$ as defined in (3), and the right-hand side is the product of the powered-sums on subvector $x_\alpha$ with weights $\mathbf{w}^\alpha$ along the same elimination order; that is, $\sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp\left[\theta_\alpha(x_\alpha)\right] = \sum_{x_{k_c}}^{w_{k_c}^\alpha} \cdots \sum_{x_{k_1}}^{w_{k_1}^\alpha} \exp\left[\theta_\alpha(x_\alpha)\right]$, where $x_\alpha = [x_{k_1}, \ldots, x_{k_c}]$ should be ranked with increasing index, consisting with the elimination order $[x_1, \ldots, x_n]$ as used in the left-hand side.*

Proof details can be found in Section E of the supplement. A key advantage of the bound (4) is that it decomposes the joint power sum on $x$ into a product of independent power sums over smaller cliques $x_\alpha$, which significantly reduces computational complexity and enables parallel computation.

### 4.1 Including Cost-shifting Variables

In order to increase the flexibility of the upper bound, we introduce a set of *cost-shifting* or *reparameterization* variables $\delta = \{\delta_i^\alpha(x_i) \mid \forall (i, \alpha), i \in \alpha\}$ on each variable-factor pair $(i, \alpha)$, which can be optimized to provide a much tighter upper bound. Note that $\Phi_{\boldsymbol{\tau}}(\theta)$ can be rewritten as,

$$\Phi_{\boldsymbol{\tau}}(\theta) = \log \sum_x^{\boldsymbol{\tau}} \exp\left[\sum_{i \in V} \sum_{\alpha \in N_i} \delta_i^\alpha(x_i) + \sum_{\alpha \in \mathcal{F}} \left(\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right)\right],$$

where $N_i = \{\alpha \mid \alpha \ni i\}$ is the set of cliques incident to $i$. Applying inequality (4), we have that

$$\Phi_{\boldsymbol{\tau}}(\theta) \leq \sum_{i \in V} \log \sum_{x_i}^{w_i} \exp\left[\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)\right] + \sum_{\alpha \in \mathcal{F}} \log \sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right] \stackrel{\text{def}}{=\!=} L(\delta, \mathbf{w}), \tag{5}$$

where the nodes $i \in V$ are also treated as cliques within inequality (4), and a new weight $w_i$ is introduced on each variable $i$; the new weights $\mathbf{w} = \{w_i, w_i^\alpha \mid \forall (i, \alpha), i \in \alpha\}$ should satisfy

$$w_i + \sum_{\alpha \in N_i} w_i^\alpha = \tau_i, \quad w_i \geq 0, \quad w_i^\alpha \geq 0, \quad \forall (i, \alpha). \tag{6}$$

The bound $L(\delta, \mathbf{w})$ is convex w.r.t. the cost-shifting variables $\delta$ and weights $\mathbf{w}$, enabling an efficient optimization algorithm that we present in Section 5. As we will discuss in Section 5.1, these shifting variables correspond to Lagrange multipliers that enforce a moment matching condition.

### 4.2 Dual Form and Connection With Existing Bounds

It is straightforward to see that our bound in (5) reduces to dual decomposition [31] when applied on MAP inference with all $\tau_i = 0$, and hence $w_i = w_i^\alpha = 0$. On the other hand, its connection with sum-inference bounds such as WMB and TRW is seen more clearly via a dual representation of (5):

**Theorem 4.2.** *The tightest upper bound obtainable by (5), that is,*

$$\min_{\mathbf{w}} \min_\delta L(\delta, \mathbf{w}) = \min_{\mathbf{w}} \max_{\mathbf{b} \in \mathbb{L}(G)} \left\{\langle \theta, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^\alpha H(x_i | x_{\mathrm{pa}_i^\alpha}; b_\alpha)\right\}, \tag{7}$$

*where $\mathbf{b} = \{b_i(x_i), b_\alpha(x_\alpha) \mid \forall (i, \alpha), i \in \alpha\}$ is a set of pseudo-marginals (or beliefs) defined on the singleton variables and the cliques, and $\mathbb{L}$ is the corresponding local consistency polytope defined by $\mathbb{L}(G) = \{\mathbf{b} \mid b_i(x_i) = \sum_{x_{\alpha \setminus i}} b_\alpha(x_\alpha), \ \sum_{x_i} b_i(x_i) = 1\}$. Here, $H(\cdot)$ are their corresponding marginal or conditional entropies, and $\mathrm{pa}_i^\alpha$ is the set of variables in $\alpha$ that rank later than $i$, that is, for the global elimination order $[x_1, \ldots, x_n]$, $\mathrm{pa}_i^\alpha = \{j \in \alpha \mid j \succ i\}$.*

The proof details can be found in Section F of the supplement. It is useful to compare Theorem 4.2 with other dual representations. As the sum of non-negatively weighted conditional entropies, the bound is clearly convex and within the general class of conditional entropy decompositions (CED) [5], but unlike generic CED it has a simple and efficient primal form (5). [4] Comparing

---

[4]The primal form derived in [5] (a geometric program) is computationally infeasible.

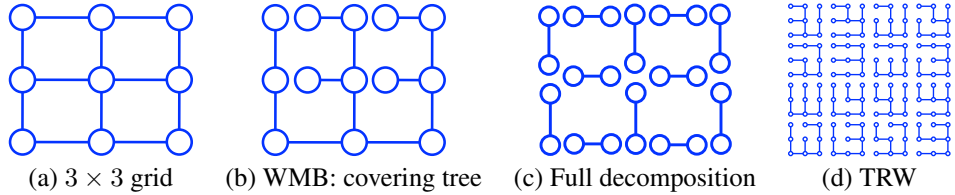|  |  |  |  |
| :---: | :---: | :---: | :---: |
| (a) $3 \times 3$ grid | (b) WMB: covering tree | (c) Full decomposition | (d) TRW |

Figure 1: Illustrating WMB, TRW and our bound on (a) $3 \times 3$ grid. (b) WMB uses a covering tree with a minimal number of splits and cost-shifting. (c) Our decomposition (5) further splits the graph into small cliques (here, edges), introducing additional cost-shifting variables but allowing for easier, monotonic optimization. (d) Primal TRW splits the graph into many spanning trees, requiring even more cost-shifting variables. Note that all three bounds attain the same tightness after optimization.

to the dual form of WMB in Theorem 4.2 of [16], our bound is as tight as WMB, and hence the class of TRW / CED bounds attainable by WMB [16]. Most duality-based forms [e.g., 9, 10] are expressed in terms of joint entropies, $\langle \theta, b \rangle + \sum_\beta c_\beta H(b_\beta)$, rather than conditional entropies; while the two can be converted, the resulting counting numbers $c_\beta$ will be differences of weights $\{w_i^\alpha\}$, [5] which obfuscates its convexity, makes it harder to maintain the relative constraints on the counting numbers during optimization, and makes some counting numbers negative (rendering some methods inapplicable [8]). Finally, like most variational bounds in dual form, the RHS of (7) has a inner maximization and hence guaranteed to bound $\Phi_\tau(\theta)$ only at its optimum.

In contrast, our Eq. (5) is a primal bound (hence, a bound for any $\delta$). It is similar to the primal form of TRW, except that (1) the individual regions are single cliques, rather than spanning trees of the graph, [6] and (2) the fraction weights $\mathbf{w}^\alpha$ associated with each region are vectors, rather than a single scalar. The representation's efficiency can be seen with an example in Figure 1, which shows a $3 \times 3$ grid model and three relaxations that achieve the same bound. Assuming $d$ states per variable and ignoring the equality constraints, our decomposition in Figure 1(c) uses $24d$ cost-shifting parameters ($\delta$), and 24 weights. WMB (Figure 1(b)) is slightly more efficient, with only $8d$ parameters for $\delta$ and and 8 weights, but its lack of decomposition makes parallel and monotonic updates difficult. On the other hand, the equivalent primal TRW uses 16 spanning trees, shown in Figure 1(d), for $16 \cdot 8 \cdot d^2$ parameters, and 16 weights. The increased dimensionality of the optimization slows convergence, and updates are non-local, requiring full message-passing sweeps on the involved trees (although this cost can be amortized in some cases [21]).

## 5 Monotonically Tightening the Bound

In this section, we propose a block coordinate descent algorithm (Algorithm 1) to minimize the upper bound $L(\delta, \mathbf{w})$ in (5) w.r.t. the shifting variables $\delta$ and weights $\mathbf{w}$. Our algorithm has a monotonic convergence property, and allows efficient, distributable local computation due to the full decomposition of our bound. Our framework allows generic powered-sum inference, including max-, sum-, or mixed-inference as special cases by setting different weights.

### 5.1 Moment Matching and Entropy Matching

We start with deriving the gradient of $L(\delta, \mathbf{w})$ w.r.t. $\delta$ and $\mathbf{w}$. We show that the zero-gradient equation w.r.t. $\delta$ has a simple form of moment matching that enforces a consistency between the *singleton beliefs* with their related *clique beliefs*, and that of weights $\mathbf{w}$ enforces a consistency of *marginal* and *conditional entropies*.

**Theorem 5.1.** *(1) For $L(\delta, \mathbf{w})$ in (5), its zero-gradient w.r.t. $\delta_i^\alpha(x_i)$ is*

$$\frac{\partial L}{\partial \delta_i^\alpha(x_i)} = \mu_i(x_i) - \sum_{x_{\alpha \setminus i}} \mu_\alpha(x_\alpha) = 0, \tag{8}$$

---

[5]See more details of this connection in Section F.3 of the supplement.

[6]While non-spanning subgraphs can be used in the primal TRW form, doing so leads to loose bounds; in contrast, our decomposition's terms consist of individual cliques.

---

**Algorithm 1** Generalized Dual-decomposition (GDD)

---

**Input:** weights $\{\tau_i \mid i \in V\}$, elimination order $\mathbf{o}$.
**Output:** the optimal $\delta^*, \mathbf{w}^*$ giving tightest upper bound $L(\delta^*, \mathbf{w}^*)$ for $\Phi_\tau(\theta)$ in (5).

initialize $\delta = 0$ and weights $\mathbf{w} = \{w_i, w_i^\alpha\}$.
**repeat**
  **for** node $i$ (in parallel with node j, $(i, j) \notin E$) **do**
    **if** $\tau_i = 0$ **then**
      update $\boldsymbol{\delta}_{N_i} = \{\delta_i^\alpha | \forall \alpha \in N_i\}$ with the closed-form update (11);
    **else if** $\tau_i \neq 0$ **then**
      update $\boldsymbol{\delta}_{N_i}$ and $\mathbf{w}_{N_i}$ with gradient descent (8) and (12), combined with line search;
    **end if**
  **end for**
**until** convergence
$\delta^* \leftarrow \delta, \mathbf{w}^* \leftarrow \mathbf{w}$, and evaluate $L(\delta^*, \mathbf{w}^*)$ by (5);

*Remark.* GDD solves max-, sum- and mixed-inference by setting different values of weights $\{\tau_i\}$.

---

where $\mu_i(x_i) \propto \exp\left[\frac{1}{w_i} \sum_{\alpha \in N_i} \delta_i^\alpha(x_i)\right]$ *can be interpreted as a singleton belief on* $x_i$, *and* $\mu_\alpha(x_\alpha)$ *can be viewed as clique belief on* $x_\alpha$, *defined with a chain rule (assuming* $x_\alpha = [x_1, \ldots, x_c]$),
$\mu_\alpha(x_\alpha) = \prod_{i=1}^c \mu_\alpha(x_i | x_{i+1:c}); \; \mu_\alpha(x_i | x_{i+1:c}) = (Z_{i-1}(x_{i:c})/Z_i(x_{i+1:c}))^{1/w_i^\alpha}$, *where* $Z_i$ *is the partial powered-sum up to* $x_{1:i}$ *on the clique, that is,*

$$Z_i(x_{i+1:c}) = \sum_{x_i}^{w_i^\alpha} \cdots \sum_{x_1}^{w_1^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right], \quad Z_0(x_\alpha) = \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right],$$

*where the summation order should be consistent with the global elimination order* $\mathbf{o} = [x_1, \ldots, x_n]$.

*(2) The gradients of* $L(\delta, \mathbf{w})$ *w.r.t. the weights* $\{w_i, w_i^\alpha\}$ *are marginal and conditional entropies defined on the beliefs* $\{\mu_i, \mu_\alpha\}$, *respectively,*

$$\frac{\partial L}{\partial w_i} = H(x_i; \mu_i), \qquad \frac{\partial L}{\partial w_i^\alpha} = H(x_i | x_{i+1:c}; \mu_\alpha) = -\sum_{x_\alpha} \mu_\alpha(x_\alpha) \log \mu_\alpha(x_i | x_{i+1:c}). \tag{9}$$

*Therefore, the optimal weights should satisfy the following KKT condition*

$$w_i\big(H(x_i; \mu_i) - \bar{H}_i\big) = 0, \quad w_i^\alpha\big(H(x_i | x_{i+1:c}; \mu_\alpha) - \bar{H}_i\big) = 0, \quad \forall(i, \alpha) \tag{10}$$

*where* $\bar{H}_i = w_i H(x_i; \mu_i) + \sum_\alpha w_i^\alpha H(x_i | x_{i+1:c}; \mu_\alpha)$ *is the (weighted) average entropy on node* $i$.

The proof details can be found in Section G of the supplement. The matching condition (8) enforces that $\mu = \{\mu_i, \mu_\alpha \mid \forall(i, \alpha)\}$ belong to the local consistency polytope $\mathbb{L}$ as defined in Theorem 4.2; similar moment matching results appear commonly in variational inference algorithms [e.g., 34]. [34] also derive a gradient of the weights, but it is based on the free energy form and is correct only after optimization; our form holds at any point, enabling efficient joint optimization of $\delta$ and $\mathbf{w}$.

## 5.2 Block Coordinate Descent

We derive a block coordinate descent method in Algorithm 1 to minimize our bound, in which we sweep through all the nodes $i$ and update each block $\boldsymbol{\delta}_{N_i} = \{\delta_i^\alpha(x_i) \mid \forall \alpha \in N_i\}$ and $\mathbf{w}_{N_i} = \{w_i, w_i^\alpha \mid \forall \alpha \in N_i\}$ with the neighborhood parameters fixed. Our algorithm applies two update types, depending on whether the variables have zero weight: (1) For nodes with $\tau_i = 0$ (corresponding to max nodes $i \in B$ in marginal MAP), we derive a closed-form coordinate descent rule for the associated shifting variables $\boldsymbol{\delta}_{N_i}$; these nodes do not require to optimize $\mathbf{w}_{N_i}$ since it is fixed to be zero. (2) For nodes with $\tau_i \neq 0$ (e.g., sum nodes $i \in A$ in marginal MAP), we lack a closed form update for $\boldsymbol{\delta}_{N_i}$ and $\mathbf{w}_{N_i}$, and optimize by local gradient descent combined with line search.

The lack of a closed form coordinate update for nodes $\tau_i \neq 0$ is mainly because the order of power sums with different weights cannot be exchanged. However, the gradient descent inner loop is still efficient, because each gradient evaluation only involves the local variables in clique $\alpha$.

**Closed-form Update.** For any node $i$ with $\tau_i = 0$ (i.e., max nodes $i \in B$ in marginal MAP), and its associated $\boldsymbol{\delta}_{N_i} = \{\delta_i^\alpha(x_i) \mid \forall \alpha \in N_i\}$, the following update gives a closed form solution for the zero (sub-)gradient equation in (8) (keeping the other $\{\delta_j^\alpha | j \neq i, \forall \alpha \in N_i\}$ fixed):

$$\delta_i^\alpha(x_i) \leftarrow \frac{|N_i|}{|N_i|+1}\gamma_i^\alpha(x_i) - \frac{1}{|N_i|+1}\sum_{\beta \in N_i \setminus \alpha} \gamma_i^\beta(x_i), \qquad (11)$$

where $|N_i|$ is the number of neighborhood cliques, and $\gamma_i^\alpha(x_i) = \log \sum_{x_{\alpha \setminus i}}^{\mathbf{w}_{\setminus i}^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{j \in \alpha \setminus i} \delta_j^\alpha(x_j)\right]$. Note that the update in (11) works regardless of the weights of nodes $\{\tau_j \mid \forall j \in \alpha, \ \forall \alpha \in N_i\}$ in the neighborhood cliques; when all the neighboring nodes also have zero weight ($\tau_j = 0$ for $\forall j \in \alpha, \ \forall \alpha \in N_i$), it is analogous to the "star" update of dual decomposition for MAP [31]. The detailed derivation is shown in Proposition H.2 in the supplement.

The update in (11) can be calculated with a cost of only $O(|N_i| \cdot d^{|\alpha|})$, where $d$ is the number of states of $x_i$, and $|\alpha|$ is the clique size, by computing and saving all the shared $\{\gamma_i^\alpha(x_i)\}$ before updating $\boldsymbol{\delta}_{N_i}$. Furthermore, the updates of $\boldsymbol{\delta}_{N_i}$ for different nodes $i$ are independent if they are not directly connected by some clique $\alpha$; this makes it easy to parallelize the coordinate descent process by partitioning the graph into independent sets, and parallelizing the updates within each set.

**Local Gradient Descent.**     For nodes with $\tau_i \neq 0$ (or $i \in A$ in marginal MAP), there is no closed-form solution for $\{\delta_i^\alpha(x_i)\}$ and $\{w_i, w_i^\alpha\}$ to minimize the upper bound. However, because of the fully decomposed form, the gradient w.r.t. $\boldsymbol{\delta}_{N_i}$ and $\mathbf{w}_{N_i}$, (8)–(9), can be evaluated efficiently via local computation with $O(|N_i| \cdot d^{|\alpha|})$, and again can be parallelized between nonadjacent nodes. To handle the normalization constraint (6) on $\mathbf{w}_{N_i}$, we use an exponential gradient descent: let $w_i = \exp(v_i)/\left[\exp(v_i) + \sum_\alpha \exp(v_i^\alpha)\right]$ and $w_i^\alpha = \exp(v_i^\alpha)/\left[\exp(v_i) + \sum_\alpha \exp(v_i^\alpha)\right]$; taking the gradient w.r.t. $v_i$ and $v_i^\alpha$ and transforming back gives the following update

$$w_i \propto w_i \exp\left[-\eta w_i\big(H(x_i; \mu_i) - \bar{H}_i\big)\right], \quad w_i^\alpha \propto w_i^\alpha \exp\left[-\eta w_i^\alpha\big(H(x_i | x_{\mathrm{pa}_i^\alpha}; \mu_\alpha) - \bar{H}_i\big)\right], \quad (12)$$

where $\eta$ is the step size and $\mathrm{pa}_i^\alpha = \{j \in \alpha \mid j \succ i\}$. In our implementation, we find that a few gradient steps (e.g., 5) with a backtracking line search using the Armijo rule works well in practice. Other more advanced optimization methods, such as L-BFGS and Newton's method are also applicable.

## 6   Experiments

In this section, we demonstrate our algorithm on a set of real-world graphical models from recent UAI inference challenges, including two diagnostic Bayesian networks with 203 and 359 variables and max domain sizes 7 and 6, respectively, and several MRFs for pedigree analysis with up to 1289 variables, max domain size of 7 and clique size 5.[7] We construct marginal MAP problems on these models by randomly selecting half of the variables to be max nodes, and the rest as sum nodes.

We implement several algorithms that optimize the same primal marginal MAP bound, including our GDD (Algorithm 1), the WMB algorithm in [16] with $ibound = 1$, which uses the same cliques and a fixed point heuristic for optimization, and an off-the-shelf L-BFGS implementation that directly optimizes our decomposed bound. For comparison, we also computed several related primal bounds, including standard mini-bucket [2] and elimination reordering [27, 38], limited to the same computational limits ($ibound = 1$). We also tried MAS [20] but found its bounds extremely loose.[8]

Decoding (finding a configuration $\hat{x}_B$) is more difficult in marginal MAP than in joint MAP. We use the same local decoding procedure that is standard in dual decomposition [31]. However, evaluating the objective $Q(\hat{x}_B)$ involves a potentially difficult sum over $x_A$, making it hard to score each decoding. For this reason, we evaluate the score of each decoding, but show the most recent decoding rather than the best (as is standard in MAP) to simulate behavior in practice.

Figure 2 and Figure 3 compare the convergence of the different algorithms, where we define the iteration of each algorithm to correspond to a full sweep over the graph, with the same order of time complexity: one iteration for GDD is defined in Algorithm 1; for WMB is a full forward and backward message pass, as in Algorithm 2 of [16]; and for L-BFGS is a joint quasi-Newton step on all variables. The elimination order that we use is obtained by a weighted-min-fill heuristic [1] constrained to eliminate the sum nodes first.

**Diagnostic Bayesian Networks.**     Figure 2(a)-(b) shows that our GDD converges quickly and monotonically on both the networks, while WMB does not converge without proper damping; we

---

[8]The instances tested have many zero probabilities, which make finding lower bounds difficult; since MAS' bounds are symmetrized, this likely contributes to its upper bounds being loose.
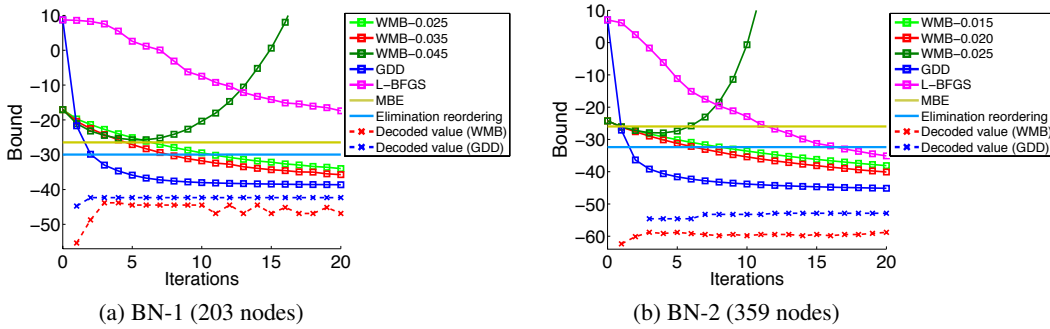
(a) BN-1 (203 nodes)

(b) BN-2 (359 nodes)

Figure 2: Marginal MAP results on BN-1 and BN-2 with $50\%$ randomly selected max-nodes (additional plots are in the supplement B). We plot the upper bounds of different algorithms across iterations; the objective function $Q(x_B)$ (2) of the decoded solutions $x_B$ are also shown (dashed lines). At the beginning, $Q(x_B)$ may equal to $-\infty$ because of zero probabiliy.



(a) pedigree1 (334 nodes)

(b) pedigree7 (1068 nodes)
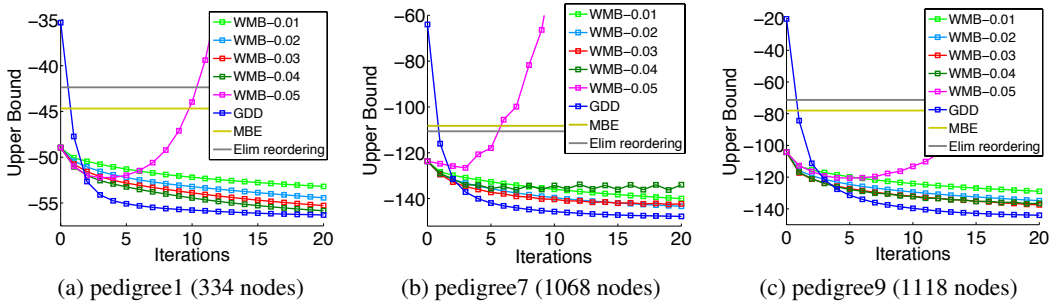
(c) pedigree9 (1118 nodes)

Figure 3: Marginal MAP inference on three pedigree models (additional plots are in the supplement C). We randomly select half the nodes as max-nodes in these models. We tune the damping rate of WMB from 0.01 to 0.05.

experimented different damping ratios for WMB, and found that it is slower than GDD even with the best damping ratio found (e.g., in Figure 2(a), WMB works best with damping ratio 0.035 (WMB-0.035), but is still significantly slower than GDD). Our GDD also gives better decoded marginal MAP solution $x_B$ (obtained by rounding the singleton beliefs). Both WMB and our GDD provide a much tighter bound than the non-iterative mini-bucket elimination (MBE) [2] or reordered elimination [27, 38] methods.

**Genetic Pedigree Instances.** Figure 3 shows similar results on a set of pedigree instances. Again, GDD outperforms WMB even with the best possible damping, and out-performs the non-iterative bounds after only one iteration (pass through the graph).

## 7 Conclusion

In this work, we propose a new class of decomposition bounds for general powered-sum inference, which is capable of representing a large class of primal variational bounds but is much more computationally efficient. Unlike previous primal sum bounds, our bound decomposes into computations on small, local cliques, increasing efficiency and enabling parallel and monotonic optimization. We derive a block coordinate descent algorithm for optimizing our bound over both the cost-shifting parameters (reparameterization) and weights (fractional counting numbers), which generalizes dual decomposition and enjoy similar monotonic convergence property. Taking the advantage of its monotonic convergence, our new algorithm can be widely applied as a building block for improved heuristic construction in search, or more efficient learning algorithms.

# References

[1] R. Dechter. Reasoning with probabilistic and deterministic graphical models: Exact algorithms. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2013.

[2] R. Dechter and I. Rish. Mini-buckets: A general scheme for bounded inference. *JACM*, 2003.

[3] J. Domke. Dual decomposition for marginal inference. In *AAAI*, 2011.

[4] A. Doucet, S. Godsill, and C. Robert. Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, 2002.

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

[6] A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2008.

[7] G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities*. Cambridge University Press, 1952.

[8] T. Hazan, J. Peng, and A. Shashua. Tightening fractional covering upper bounds on the partition function for high-order region graphs. In *UAI*, 2012.

[9] T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *UAI*, 2008.

[10] T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 2010.

[11] A. Ihler, N. Flerova, R. Dechter, and L. Otten. Join-graph based cost-shifting schemes. In *UAI*, 2012.

[12] J. Jancsary and G. Matz. Convergent decomposition solvers for TRW free energies. In *AISTATS*, 2011.

[13] I. Kiselev and P. Poupart. Policy optimization by marginal MAP probabilistic inference in generative models. In *AAMAS*, 2014.

[14] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *TPAMI*, 2011.

[15] Q. Liu. *Reasoning and Decisions in Probabilistic Graphical Models–A Unified Framework*. PhD thesis, University of California, Irvine, 2014.

[16] Q. Liu and A. Ihler. Bounding the partition function using Hölder's inequality. In *ICML*, 2011.

[17] Q. Liu and A. Ihler. Variational algorithms for marginal MAP. *JMLR*, 2013.

[18] R. Marinescu, R. Dechter, and A. Ihler. AND/OR search for marginal MAP. In *UAI*, 2014.

[19] D. Maua and C. de Campos. Anytime marginal maximum a posteriori inference. In *ICML*, 2012.

[20] C. Meek and Y. Wexler. Approximating max-sum-product problems using multiplicative error bounds. *Bayesian Statistics*, 2011.

[21] T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms: a unifying view. In *UAI*, 2009.

[22] O. Meshi and A. Globerson. An alternating direction method for dual MAP LP relaxation. In *ECML/PKDD*, 2011.

[23] O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson. Learning efficiently with approximate inference via dual losses. In *ICML*, 2010.

[24] J. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 2010.

[25] J. Naradowsky, S. Riedel, and D. Smith. Improving NLP through marginalization of hidden syntactic structure. In *EMNLP*, 2012.

[26] S. Nowozin and C. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6, 2011.

[27] J. Park and A. Darwiche. Solving MAP exactly using systematic search. In *UAI*, 2003.

[28] J. Park and A. Darwiche. Complexity results and approximation strategies for MAP explanations. *JAIR*, 2004.

[29] W. Ping, Q. Liu, and A. Ihler. Marginal structured SVM with hidden variables. In *ICML*, 2014.

[30] N. Ruozzi and S. Tatikonda. Message-passing algorithms: Reparameterizations and splittings. *IEEE Transactions on Information Theory*, 2013.

[31] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. *Optimization for Machine Learning*, 2011.

[32] D. Sontag and T. Jaakkola. Tree block coordinate descent for MAP in graphical models. *AISTATS*, 2009.

[33] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

[34] M. Wainwright, T. Jaakkola, and A. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 2005.

[35] Y. Weiss, C. Yanover, and T. Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.

[36] T. Werner. A linear programming approach to max-sum problem: A review. *TPAMI*, 2007.

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

[38] C. Yuan and E. Hansen. Efficient computation of jointree bounds for systematic map search. *IJCAI*, 2009.

[39] C. Yuan, T. Lu, and M. Druzdzel. Annealed MAP. In *UAI*, 2004.

# Supplement: Decomposition Bounds for Marginal MAP

**Wei Ping**[*]     **Qiang Liu**[†]     **Alexander Ihler**[*]
[*]Computer Science, UC Irvine     [†]Computer Science, Dartmouth College
{wping,ihler}@ics.uci.edu    qliu@cs.dartmouth.edu

## A    Experiment on Ising grid

Our GDD directly optimizes a primal bound, and is thus guaranteed to be an upper bound of the partition function even before the algorithm converges, enabling a desirable "any-time" property. In contrast, typical implementations of tree reweighted (TRW) belief propagation optimize the dual free energy function [4], and are not guaranteed to be a bound before convergence. We illustrate this point using an experiment on a toy $5 \times 5$ Ising grid, with parameters generated by normal ditribution $N(0, 2)$ and half nodes selected as max-nodes for marginal MAP. Figure 1(a)-(b) shows the TRW free energy objective and GDD, WMB upper bounds across iterations; we can see that TRW does violate the upper bound property before convergence, while GDD and WMB always give valid upper bounds.
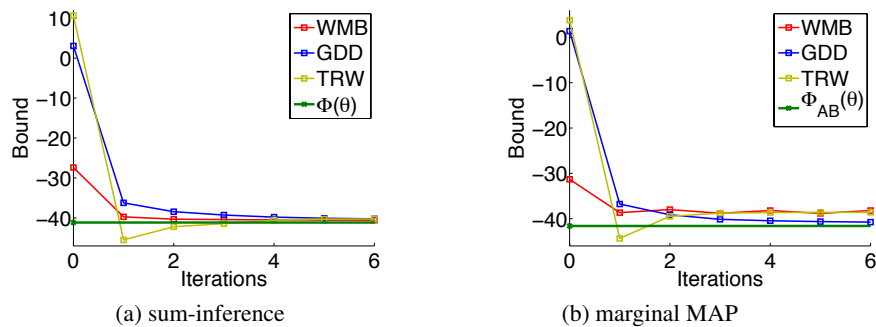


(a) sum-inference             (b) marginal MAP

Figure 1: Sum-inference and marginal MAP results on a toy Ising model ($5 \times 5$ grid). Each iteration of the different algorithms corresponds to a full sweep over the graph. Note that the dual formulation (TRW) is not a bound until convergence; for example, at iteration 1, its objective function is below the true $\Phi$.

## B    More Results on Diagnostic Bayesian Networks

In addtion to the marginal MAP results on BN-1 and BN-2 in main text, we vary the percentage of max-nodes when generating the marginal MAP problems; the reported results in Figure 2(a)-(b) are the best bound obtained by the different algorithms with the first 20 iterations. In all cases, GDD's results are as good or better than WMB. WMB-0.5 (WMB with damping ratio $0.5$) appears to work well on sum-only and max-only (MAP) problems, i.e., when the percentage of max-nodes equals $0\%$ and $100\%$ respectively, but performs very poorly on intermediate settings. The far more heavily damped WMB-0.04 or WMB-0.02 work better on average, but have much slower convergence.

## C    More Results on Pedigree Linkage Analysis

We test our algorithm on additional 6 models of pedigree linkage analysis from the UAI08 inference challenge. We construct marginal MAP problems by randomly selected $50\%$ of nodes to be max-
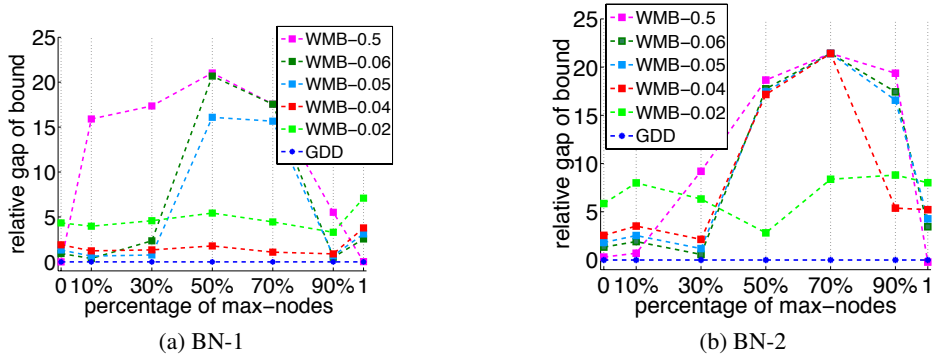
Figure 2: More marginal MAP results (including sum-inference and MAP) on two diagnostic Bayesian networks. We report the best results obtained by GDD and WMB with 20 iterations in marginal MAP problems constructed by randomly selecting different percentages of max-nodes.
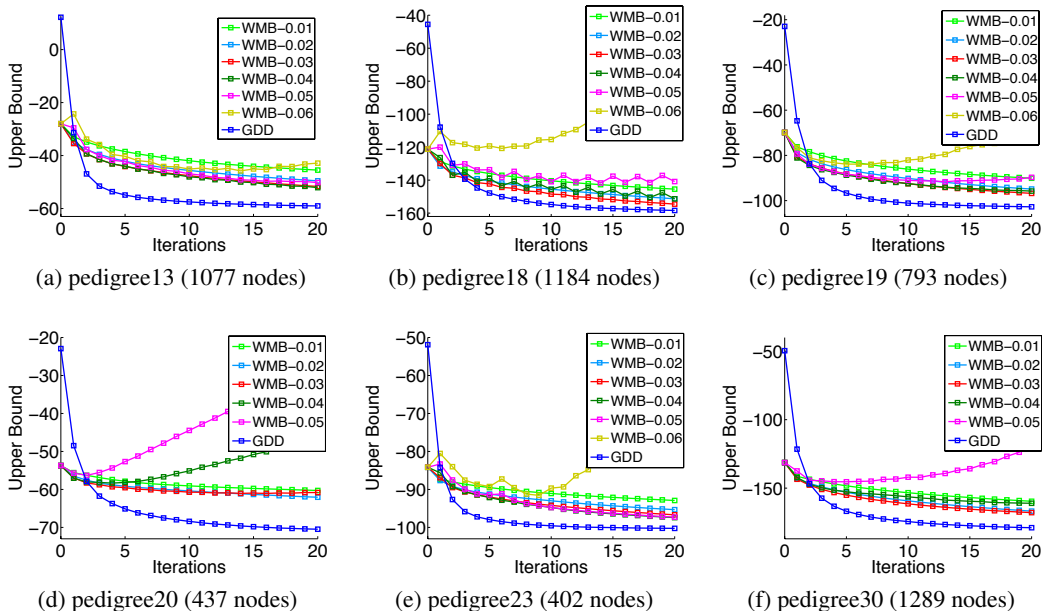


Figure 3: Marginal MAP inference on additional pedigree linkage analysis models. We randomly selected $50\%$ of nodes as max-nodes in these models. We tune the damping rate of WMB from 0.01 to 0.06, but we omit WMB-0.06 in the plot if WMB-0.05 is already diverged.

nodes, and report all the results in Figure 3. We find that our algorithm consistently outperforms WMB with the best possible damping ratio.

## D  Extensions to Junction Graph

Our bound (5) in main text uses a standard "factor graph" representation in which the cost-shifts $\{\delta_i^\alpha\}$ are defined for each variable-factor pair $(i, \alpha)$, and are functions of single variables $x_i$. We can extend our bound to use more general shifting parameters using a junction graph representation; this allows us to exploit higher order clique structures, leading to better performance.

Let $(\mathcal{C}, \mathcal{S})$ be a junction graph of $p(x; \theta)$ where $\mathcal{C} = \{c \mid c \subset V\}$ is the set of clusters, and $\mathcal{S} = \{s = c_k \cap c_l \mid c_k, c_l \in \mathcal{C}\}$ is the set of separators. Assume $p(x; \theta)$ can be reparameterized into the form,

$$p(x; \theta) = \exp\left[\sum_{c \in \mathcal{C}} \theta_c(x_c) - \Phi(\theta)\right], \tag{1}$$

2

and the weighted log partition function is rewritten as $\Phi_{\tau}(\theta) = \log \sum_{x}^{\tau} \exp \left[ \sum_{c \in \mathcal{C}} \theta_c(x_c) \right]$. Similar to the derivation of bound (5) in main text, we can apply Theorem 4.1, but with a set of more general cost-shifting variables $\delta_s^c$, defined on each adjacent separator-cluster pair $(s, c)$; this gives the more general upper bound,

$$\Phi_{\tau}(\theta) \leq \sum_{s \in \mathcal{S}} \log \sum_{x_s}^{\mathbf{w}^s} \exp \left[ \sum_{c \supseteq s} \delta_s^c(x_s) \right] + \sum_{c \in \mathcal{C}} \log \sum_{x_c}^{\mathbf{w}^c} \exp \left[ \theta_c(x_c) - \sum_{s \subseteq c} \delta_s^c(x_s) \right], \qquad (2)$$

where we introduce the set of non-negative weights $\mathbf{w}^s = \{w_i^s \mid i \in s\}$ on each separator and $\mathbf{w}^c = \{w_i^c \mid i \in c\}$ on each cluster, which should satisfy $\sum_{s \in N_i^{se}} w_i^s + \sum_{c \in N_i^c} w_i^c = \tau_i$, where $N_i^{se} = \{s \mid i \in s\}$ are all the separators that include node $i$, and $N_i^c = \{c \mid i \in c\}$ are all the clusters that include node $i$. Obviously, our earlier bound (5) in main text can be viewed as a special case of (2) with a special junction graph whose separators consist of only single variables, that is, $\mathcal{S} = V$.

A block coordinate descent algorithm similar to Algorithm 1 can be derived to optimize the junction graph bound. In this case, we sweep through all the separators $s$ and perform block coordinate update on all $\{\delta_s^c | \forall c \supseteq s\}$ at each iteration. Similarly to Algorithm 1, we can derive a close form update for separators with all-zero weights (that is, $\tau_i = 0$, $\forall i \in s$, corresponding to $s \subseteq B$ in marginal MAP), and perform local gradient descent otherwise.

# E Proof of Thereom 4.1

*Proof.* Note the Hölder's inequality is

$$\left[ \sum_x \prod_j f_j(x)^{1/\xi_0} \right]^{\xi_0} \leq \prod_j \left[ \sum_x f_j(x)^{1/\xi_j} \right]^{\xi_j},$$

where $\{f_j(x)\}$ are arbitrary positive functions, and $\{\xi_j\}$ are non-negative numbers that satisfy $\sum_j \xi_j = \xi_0$. Note we extend the inequality by defining power sum with $\xi_j = 0$ to equal the max operator. Our result follows by applying Hölder's inequality on each $x_i$ sequentially along the elimination order $[x_1, x_2, \cdots, x_n]$. $\qquad \square$

# F Dual Representations

## F.1 Background

The log-partition function $\Phi(\theta)$ has the following variational (dual) form

$$\Phi(\theta) = \log \sum_x \exp(\theta(x)) = \max_{b \in \mathbb{M}(G)} \left\{ \langle \theta, b \rangle + H(x; b) \right\}$$

where $\mathbb{M}(G)$ is the *marginal polytope* [5]. Then, for any scalar $\varepsilon > 0$ (including $\varepsilon \to 0^+$), we have

$$\Phi_{\varepsilon}(\theta) = \varepsilon \log \sum_x \exp\left(\frac{\theta(x)}{\varepsilon}\right) = \varepsilon \max_{b \in \mathbb{M}} \left\{ \langle \frac{\theta}{\varepsilon}, b \rangle + H(x; b) \right\} = \max_{b \in \mathbb{M}} \left\{ \langle \theta, b \rangle + \varepsilon H(x; b) \right\}.$$

As stated in [2, 3], we can further generalize the variational form of above scalar-weighted log partition function to the vector-weighted log partition function (3) in the main text,

$$\Phi_{\tau}(\theta) = \log \sum_{x_n}^{\tau_n} \cdots \sum_{x_1}^{\tau_1} \exp(\theta(x)) = \max_{b \in \mathbb{M}(G)} \left\{ \langle \theta, b \rangle + \sum_i \tau_i H(x_i | x_{i+1:n}; b) \right\}, \qquad (3)$$

where $H(x_i | x_{i+1:n}; b)$ is the conditional entropy on $b(x)$, and is defined as $H(x_i | x_{i+1:n}; b) = -\sum_x b(x) \log(b(x_i | x_{i+1:n}))$. See more details of its derivation in Theorem 4.1 within [2].

## F.2 Proof of Thereom 4.2

We will prove the following dual representation of our bound,

$$\min_{\delta} L(\delta, \mathbf{w}) = \max_{\mathbf{b} \in \mathbb{L}(G)} \left\{ \langle \theta, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^{\alpha} H(x_i | x_{\mathrm{pa}_i^{\alpha}}; b_{\alpha}) \right\}, \qquad (4)$$

3

where $\mathbb{L}(G) = \{\mathbf{b} \mid b_i(x_i) = \sum_{x_{\alpha \setminus i}} b_\alpha(x_\alpha), \ \sum_{x_i} b_i(x_i) = 1\}$ is the local consistency polytope, and $\text{pa}_i^\alpha = \{j \in \alpha \mid j \succ i\}$. Thereom 4.2 follows directly from (4).

*Proof.* In our primal bound $L(\delta, \mathbf{w})$ (5) in main text, we let $\widetilde{\theta}_i(x_i) = \theta_i(x_i) + \sum_{\alpha \in N_i} \delta_i^\alpha(x_i)$ (we add dummy singleton $\theta_i(x_i) \equiv 0$), and $\widetilde{\theta}_\alpha(x_\alpha) = \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)$, then the bound can be rewritten as,

$$L(\widetilde{\theta}, \mathbf{w}) = \sum_{i \in V} \log \sum_{x_i}^{w_i} \exp\left[\widetilde{\theta}_i(x_i)\right] + \sum_{\alpha \in \mathcal{F}} \log \sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp\left[\widetilde{\theta}_\alpha(x_\alpha)\right].$$

Note, for any assignment $x$, we have $\sum_i \widetilde{\theta}_i(x_i) + \sum_\alpha \widetilde{\theta}_\alpha(x_\alpha) = \sum_\alpha \theta_\alpha(x_\alpha)$.

By applying the dual form of the powered sum (3) on each node and clique respectively, we have

$$L(\widetilde{\theta}, \mathbf{w}) = \sum_{i \in V} \max_{b_i \in \mathbb{M}(G_i)} \left\{\langle \widetilde{\theta}_i, b_i \rangle + w_i H(x_i; b_i)\right\} + \sum_{\alpha \in \mathcal{F}} \max_{b_\alpha \in \mathbb{M}(G_\alpha)} \left\{\langle \widetilde{\theta}_\alpha, b_\alpha \rangle + \sum_{i \in \alpha} w_i^\alpha H(x_i | x_{\text{pa}_i^\alpha}; b_\alpha)\right\},$$

where $\text{pa}_i^\alpha$ is the set of variables in $\alpha$ that are summed out later than $i$, $\mathbb{M}(G_i)$ and $\mathbb{M}(G_\alpha)$ are the marginal polytopes on singleton node $i$ and clique $\alpha$ respectively, which enforce $\{b_i, b_\alpha\}$ to be properly normalized. We denote them jointly as $\widehat{\mathbb{M}} = \{\mathbb{M}(G_i), \mathbb{M}(G_\alpha) \mid \forall i \in V, \alpha \in \mathcal{F}\}$, then

$$L(\widetilde{\theta}, \mathbf{w}) = \max_{\mathbf{b} \in \widehat{\mathbb{M}}} \left\{\langle \widetilde{\theta}, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^\alpha H(x_i | x_{\text{pa}_i^\alpha}; b_\alpha)\right\},$$

where $b_i, b_\alpha \in \widehat{\mathbb{M}}$ are independently optimized.

Then, by tightening *reparameterization* $\widetilde{\theta} = \{\widetilde{\theta}_i, \widetilde{\theta}_\alpha\}$, we have

$$\min_{\widetilde{\theta}} L(\widetilde{\theta}, \mathbf{w}) = \max_{\mathbf{b} \in \widehat{\mathbb{M}}} \min_{\widetilde{\theta}} \left\{\langle \widetilde{\theta}, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^\alpha H(x_i | x_{\text{pa}_i^\alpha}; b_\alpha)\right\}$$

where the order of $\min$ and $\max$ are commuted according to the strong duality (it's convex with $\widetilde{\theta}$, and concave with $\mathbf{b}$).

The inner minimization $\min_{\widetilde{\theta}} \langle \widetilde{\theta}, b \rangle$ is a linear program, and it turns out can be solved analytically. To see this, given the relationship between $\widetilde{\theta}$ and $\theta$, we rewrite the linear program as

$$\min_{\widetilde{\theta}} \langle \widetilde{\theta}, b \rangle = \min_\delta \left\{\langle \theta, b \rangle + \sum_{i \in V} \sum_{x_i} \sum_{\alpha \in N_i} \delta_i^\alpha(x_i) b_i(x_i) - \sum_{\alpha \in \mathcal{F}} \sum_{x_\alpha} \sum_{i \in \alpha} \delta_i^\alpha(x_i) b_\alpha(x_\alpha)\right\},$$

$$= \min_\delta \left\{\langle \theta, b \rangle + \sum_{(i,\alpha)} \sum_{x_i} \delta_i^\alpha(x_i) \Big(b_i(x_i) - \sum_{x_{\alpha \setminus i}} b_\alpha(x_\alpha)\Big)\right\}.$$

Then, it is easy to observe that the linear program is either equal to $\langle \theta, b \rangle$ only if $b$ satisfy the marginalization constraint $\sum_{x_{\alpha \setminus i}} b_\alpha(x_\alpha) = b_i(x_i)$ for $\forall (i, \alpha)$, or it will become negative infinity. Considering the outer maximization, we have

$$\min_{\widetilde{\theta}} L(\widetilde{\theta}, \mathbf{w}) = \max_{\mathbf{b} \in \mathbb{L}(G)} \left\{\langle \theta, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^\alpha H(x_i | x_{\text{pa}_i^\alpha}; b_\alpha)\right\},$$

where $\mathbb{L}(G)$ is the local consistency polytope that is obtained by enforcing both $\widehat{\mathbb{M}}$ and the marginalization constraint. $\qquad \square$

### F.3 Connection with Existing Free Energy Forms

Most variational forms are expresssed in the following linear combination of local entropies [6, 1],

$$\langle \theta, b \rangle + \sum_\beta c_\beta H(b_\beta), \tag{5}$$

where $\beta$ refers the region, $c_\beta$ refers the general counting number, $b_\beta(x_\beta)$ is the local belief.

We can rewrite our dual representations (4) as,

$$\langle \theta, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \sum_{i \in \alpha} w_i^\alpha \big( H(x_i, x_{\mathrm{pa}_i^\alpha} ; b_\alpha) - H(x_{\mathrm{pa}_i^\alpha} ; b_\alpha) \big),$$

where $\mathrm{pa}_i^\alpha$ is the set of variables in $\alpha$ that rank later than $i$. Without loss of generality, assuming $x_\alpha = [x_1, \cdots, x_i, x_j, \cdots x_c]$, i.e. $x_i$ and $x_j$ are adjacent in the order, we can get

$$\langle \theta, b \rangle + \sum_{i \in V} w_i H(x_i; b_i) + \sum_{\alpha \in \mathcal{F}} \Big\{ w_1^\alpha H(x_\alpha; b_\alpha) + \sum_{[i,j] \sqsubseteq \alpha} (w_j^\alpha - w_i^\alpha) H(x_{\mathrm{pa}_i^\alpha} ; b_{\mathrm{pa}_i^\alpha}) \Big\} \quad (6)$$

where belief $b_{\mathrm{pa}_i^\alpha}$ is defined by $b_{\mathrm{pa}_i^\alpha}(x_{\mathrm{pa}_i^\alpha}) = \sum_{x_{\alpha \setminus \mathrm{pa}_i^\alpha}} b_\alpha(x_\alpha)$.

One can view (6) in terms of (5), by selecting the region $\beta \in \{i \in V\} \cup \{\alpha \in \mathcal{F}\} \cup \{\mathrm{pa}_i^\alpha \mid \forall(i, \alpha)\}$; some counting numbers $c_\beta$ will be the differences of weights $w_j^\alpha - w_i^\alpha$.

### F.4  Matching Our Bound to WMB

After the weights are optimized, our GDD bound matches to WMB bound with optimum weights. A simple weight initialization method matches our bound to WMB with uniform weights on each mini-bucket, which often gives satisfactory result; a similar procedure can be used to match the bound with more general weights as in Section D. We first set $w_i = 0$ for all nodes $i$. We then visit the nodes $x_i$ along the elimination order $\mathbf{o} = [x_1, x_2, \cdots, x_n]$, and divide $x_i$'s neighborhood cliques $N_i = \{\alpha | \alpha \ni i\}$ into two groups: (1) the *children cliques* in which all $x_{\alpha \setminus i}$ have already been eliminated, that is, $N_i^{ch} = \{\alpha \mid \forall j \in \alpha \setminus i, \; j \prec i \text{ in } \mathbf{o}\}$; (2) the other, *parent cliques* $N_i^{pa} = \{\alpha \mid \exists j \in \alpha \setminus i, \; j \succ i \text{ in } \mathbf{o}\}$. We set $w_i^\alpha = 0$ for all the children cliques ($\alpha \in N_i^{ch}$), and uniformly split the weights, that is, $w_i^\alpha = \tau_i / |N_i^{pa}|$, across the parent cliques.

## G  Proof of Therom 5.1

*Proof.* For each $\delta_i^\alpha(x_i)$, the involved terms in $L(\delta, \mathbf{w})$ are $L_i^\alpha(\delta) = \Phi_{w_i}(\delta) + \Phi_{\mathbf{w}^\alpha}(\delta)$, where

$$\Phi_{w_i}(\delta) = \log \sum_{x_i}^{w_i} \exp \Big[ \sum_{\alpha \in N_i} \delta_i^\alpha(x_i) \Big], \quad \Phi_{\mathbf{w}^\alpha}(\delta) = \log \sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp \Big[ \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i) \Big].$$

Our result follows by showing that

$$\frac{\partial \Phi_{w_i}(\delta)}{\partial \delta_i^\alpha(x_i)} = \mu_i(x_i) \qquad \text{and} \qquad \frac{\partial \Phi_{w_i}(\delta)}{w_i} = H(x_i; \mu_i),$$

$$\frac{\partial \Phi_{\mathbf{w}^\alpha}(\delta)}{\partial \delta_i^\alpha(x_i)} = -\sum_{x_{\alpha \setminus i}} \mu_\alpha(x_\alpha) \quad \text{and} \quad \frac{\partial \Phi_{\mathbf{w}^\alpha}(\delta)}{\partial w_i^\alpha} = H(x_i | x_{i+1:c}; \mu_\alpha).$$

The gradient of $\Phi_{w_i}(\delta)$ is straightforward to calculate,

$$\frac{\partial \Phi_{w_i}}{\partial \delta_i^\alpha(x_i)} = \frac{\partial}{\partial \delta_i^\alpha(x_i)} \Big( w_i \log \sum_{x_i} \exp \Big[ \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} \Big] \Big) = \frac{\exp \Big[ \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} \Big]}{Z_{w_i}} = \mu_i(x_i),$$

where $Z_{w_i} = \sum_{x_i} \exp \Big[ \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} \Big]$, and

$$\frac{\partial \Phi_{w_i}}{\partial w_i} = \log Z_{w_i} + w_i \cdot \frac{1}{Z_{w_i}} \cdot \sum_{x_i} \Big\{ \exp \Big[ \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} \Big] \cdot \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{-w_i^2} \Big\}$$

$$= \log Z_{w_i} - \sum_{x_i} \Big\{ \mu_i(x_i) \cdot \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} \Big\}$$

$$= -\sum_{x_i} \Big\{ \mu_i(x_i) \cdot \Big[ \frac{\sum_{\alpha \in N_i} \delta_i^\alpha(x_i)}{w_i} - \log Z_{w_i} \Big] \Big\} = H(x_i; \mu_i).$$

The gradient of $\Phi_{\mathbf{w}^\alpha}(\delta)$ is more involved; see Proposition I.1 for a detailed derivation. $\qquad \square$

5

# H  Derivations of Closed-form Update

We first derive the closed-form update rule for $\delta_i^\alpha(x_i)$ in Proposition H.1. We derive the closed-form update rule for the block $\boldsymbol{\delta}_{N_i} = \{\delta_i^\alpha(x_i) \mid \forall \alpha \in N_i\}$ in Proposition H.2.

**Proposition H.1.** *Given max node $i$ in marginal MAP (i.e., $\tau_i = 0$ ) and one clique $\alpha \ni i$ (i.e. $i \in \alpha$), keeping all $\delta$ fixed except $\delta_i^\alpha(x_i)$, there is a closed-form update rule,*

$$\delta_i^\alpha(x_i) \leftarrow \frac{1}{2} \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{j \in \alpha \setminus i} \delta_j^\alpha(x_j) \right] - \frac{1}{2} \sum_{\beta \in N_i \setminus \alpha} \delta_i^\beta(x_i), \tag{7}$$

*where $x_{\alpha \setminus i} = \{x_j : j \in \alpha, j \neq i\}$, $w_{\setminus i}^\alpha = \{w_j^\alpha : j \in \alpha, j \neq i\}$, and $N_i = \{\alpha \mid \alpha \ni i\}$ is the set of all clique factors in the neighborhood of node $i$. Futhermore, this update will monotonically decrease the upper bound.*

*Proof.* The terms within the bound $L(\delta, \mathbf{w})$ that depend on $\delta_i^\alpha(x_i)$ are,

$$\max_{x_i} \left[ \sum_{\alpha \in N_i} \delta_i^\alpha(x_i) \right] + \max_{x_i} \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i) \right] \tag{8}$$

The sub-gradient of (8) w.r.t. $\delta_i^\alpha(x_i)$ equal to zero if and only if,

$$x_i^* = \operatorname*{argmax}_{x_i} \left[ \sum_{\alpha \in N_i} \delta_i^\alpha(x_i) \right] = \operatorname*{argmax}_{x_i} \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i) \right],$$

which is "argmax" matching. One sufficient condition of this matching is,

$$\sum_{\alpha \in N_i} \delta_i^\alpha(x_i) = \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i) \right]$$

which impllies matching of "pseudo marginals". Then, one can pull $\delta_i^\alpha(x_i)$ outside from the operator $\log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp$, and get the closed-form equation

$$\delta_i^\alpha(x_i) = \frac{1}{2} \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{j \in \alpha \setminus i} \delta_j^\alpha(x_j) \right] - \frac{1}{2} \sum_{\beta \in N_i \setminus \alpha} \delta_i^\beta(x_i).$$

To prove monotonicity, we substitute above update equation of $\delta_i^\alpha(x_i)$ into (8); then we get,

$$\max_{x_i} \left\{ \sum_{\beta \in N_i \setminus \alpha} \delta_i^\beta(x_i) + \log \sum_{x_{\alpha \setminus i}}^{w_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{j \in \alpha \setminus i} \delta_j^\alpha(x_j) \right] \right\}. \tag{9}$$

Clearly, (9) $\leq$ (8) by using the fact that $\max_x [f(x) + g(x)] \leq \max_x f(x) + \max_x g(x)$. $\qquad \square$

**Proposition H.2.** *Given node $i \in B$ (i.e., a max node) and all neighborhood cliques $N_i = \{\alpha \mid \alpha \ni i\}$, we can jointly optimize $\boldsymbol{\delta}_{N_i} = \{\delta_i^\alpha(x_i) \mid \forall \alpha \in N_i\}$ in closed-form by keeping the other $\{\delta_j^\alpha \mid j \neq i, \forall \alpha \in N_i\}$ fixed. The update rule is,*

$$\delta_i^\alpha(x_i) \leftarrow \frac{|N_i|}{|N_i| + 1} \gamma_i^\alpha(x_i) - \frac{1}{|N_i| + 1} \sum_{\beta \in N_i \setminus \alpha} \gamma_i^\beta(x_i), \tag{10}$$

*where $|N_i|$ is the number of neighborhood cliques, and $\{\gamma_i^\alpha(x_i) \mid \forall \alpha \in N_i\}$ are defined by*

$$\gamma_i^\alpha(x_i) = \log \sum_{x_{\alpha \setminus i}}^{\mathbf{w}_{\setminus i}^\alpha} \exp \left[ \theta_\alpha(x_\alpha) - \sum_{j \in \alpha \setminus i} \delta_j^\alpha(x_j) \right]. \tag{11}$$

*Futhermore, this upate will monotonically decrease the upper bound.*

*Proof.* For $\forall \alpha \in N_i$, we have closed-form solutions for $\delta_i^\alpha(x_i)$ as Proposition H.1. We rewrite it as,

$$\forall \alpha \in N_i, \quad 2\delta_i^\alpha(x_i) + \sum_{\beta \in N_i \backslash \alpha} \delta_i^\beta(x_i) = \log \sum_{x_{\alpha \backslash i}}^{w_{\backslash i}^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{j \in \alpha \backslash i} \delta_j^\alpha(x_j)\right]. \tag{12}$$

Note, for $\forall \alpha, \beta \in N_i$, there is a linear relationship between $\delta_i^\alpha(x_i)$ and $\delta_i^\beta(x_i)$.

We denote column vector $\boldsymbol{\gamma}_i(x_i)$ filled $\alpha$-th element with

$$\gamma_i^\alpha(x_i) = \log \sum_{x_{\alpha \backslash i}}^{w_{\backslash i}^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{j \in \alpha \backslash i} \delta_j^\alpha(x_j)\right].$$

We also frame all $\{\delta_i^\alpha(x_i) \mid \alpha \in N_i\}$ into a column vector $\boldsymbol{\delta}_{N_i}(x_i)$, and denote $|N_i| \times |N_i|$ matrix A

$$A = \begin{pmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 2 \end{pmatrix}, \text{ and note } A^{-1} = \begin{pmatrix} \frac{|N_i|}{|N_i|+1} & -\frac{1}{|N_i|+1} & \cdots & -\frac{1}{|N_i|+1} \\ -\frac{1}{|N_i|+1} & \frac{|N_i|}{|N_i|+1} & \cdots & -\frac{1}{|N_i|+1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{|N_i|+1} & -\frac{1}{|N_i|+1} & \cdots & \frac{|N_i|}{|N_i|+1} \end{pmatrix}.$$

It is easy to verify $A\boldsymbol{\delta}_{N_i}(x_i) = \boldsymbol{\gamma}_i(x_i)$. from (12). Since $A$ is invertable, one can solve

$$\boldsymbol{\delta}_{N_i}(x_i) = A^{-1}\boldsymbol{\gamma}_i(x_i).$$

Then, one can read out the closed-form update rule (10). The monotonicity holds directly by noticing that the update rule (10) are solutions which jointly satisfy equation (7). $\quad\square$

## I Derivations of Gradient

**Proposition I.1.** *Given a weight vector* $\mathbf{w}^\alpha = [w_1^\alpha, \cdots, w_i^\alpha, \cdots, w_c^\alpha]$ *associated with variables* $x_\alpha = \{x_1, \cdots, x_i, \cdots, x_c\}$ *on clique* $\alpha$, *where* $c = |\alpha|$ *the power sum over clique* $\alpha$ *is,*

$$\Phi_{\mathbf{w}^\alpha}(\delta) = \log \sum_{x_\alpha}^{\mathbf{w}^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right] = \log \sum_{x_c}^{w_c^\alpha} \cdots \sum_{x_i}^{w_i^\alpha} \cdots \sum_{x_1}^{w_1^\alpha} \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right].$$

*We recursively denote* $Z_i$ *as the partial power sum up to* $x_{1:i}$,

$$Z_0(x_\alpha) = \exp\left[\theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)\right] \quad \text{and} \quad Z_i(x_{i+1:c}) = \sum_{x_i}^{w_i^\alpha} Z_{i-1}(x_{i:c}), \tag{13}$$

*thus* $\log Z_c = \Phi_{\mathbf{w}^\alpha}$. *We also denote the "pseudo marginal" (or, belief) on* $x_\alpha$,

$$\mu_\alpha(x_\alpha) = \prod_{i=1}^c \mu_\alpha(x_i|x_{i+1:c}); \quad \mu_\alpha(x_i|x_{i+1:c}) = \left(\frac{Z_{i-1}(x_{i:c})}{Z_i(x_{i+1:c})}\right)^{1/w_i^\alpha},$$

*and it is easy to verify that* $\mu_\alpha(x_i|x_{i+1:c})$ *and* $\mu_\alpha(x_\alpha)$ *are normalized.*

*Then, the derivative of* $\Phi_{\mathbf{w}^\alpha}$ *w.r.t.* $\delta_i^\alpha(x_i)$ *can be written by beliefs,*

$$\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial \delta_i^\alpha(x_i)} = -\mu_\alpha(x_i) = -\sum_{x_{\alpha \backslash i}} \mu_\alpha(x_\alpha) = -\sum_{x_c} \cdots \sum_{x_{i+1}} \prod_{j=i}^c \mu_\alpha(x_j|x_{j+1:c}) \tag{14}$$

*In addition, the derivative of* $\Phi_{\mathbf{w}^\alpha}$ *w.r.t.* $w_i^\alpha$ *is the conditional entropy,*

$$\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial w_i^\alpha} = H(x_i|x_{i+1:c}; \mu_\alpha(x_\alpha)) = -\sum_{x_\alpha} \mu_\alpha(x_\alpha) \log \mu_\alpha(x_i|x_{i+1:c}) \tag{15}$$

*Proof.*

Denote the reparameterization on clique $\alpha$ as $\widetilde{\theta}_\alpha(x_\alpha) = \theta_\alpha(x_\alpha) - \sum_{i \in \alpha} \delta_i^\alpha(x_i)$.

From the recursive definition of $Z_i(x_{i+1:c})$ (13), we have the following recursive rule for gradient,

$$
\begin{aligned}
\frac{\partial \log Z_i(x_{i+1:c})}{\partial \widetilde{\theta}_\alpha(x_\alpha)} &= \frac{\partial}{\partial \widetilde{\theta}_\alpha(x_\alpha)} \left( w_i^\alpha \log \sum_{x_i} \left[ Z_{i-1}(x_{i:c}) \right]^{1/w_i^\alpha} \right) \\
&= w_i^\alpha \cdot \frac{\frac{1}{w_i^\alpha} \cdot Z_{i-1}(x_{i:c})^{\frac{1}{w_i^\alpha}}}{\sum_{x_i} \left[ Z_{i-1}(x_{i:c}) \right]^{\frac{1}{w_c^\alpha}}} \cdot Z_{i-1}(x_{i:c})^{-1} \cdot \frac{\partial Z_{i-1}(x_{i:c})}{\partial \widetilde{\theta}_\alpha(x_\alpha)} \\
&= \frac{Z_{i-1}(x_{i:c})^{\frac{1}{w_i^\alpha}}}{\sum_{x_i} \left[ Z_{i-1}(x_{i:c}) \right]^{\frac{1}{w_c^\alpha}}} \cdot \frac{\partial \log Z_{i-1}(x_{i:c})}{\partial \widetilde{\theta}_\alpha(x_\alpha)} \\
&= \mu_\alpha(x_i | x_{i+1:c}) \cdot \frac{\partial \log Z_{i-1}(x_{i:c})}{\partial \widetilde{\theta}_\alpha(x_\alpha)}. \qquad (16)
\end{aligned}
$$

It should be noted, implicitly, $x_{i+1:c}$ within $\widetilde{\theta}_\alpha(x_\alpha)$ should take the same value as $x_{i+1:c}$ in $\log Z_i(x_{i+1:c})$, otherwise, the derivative will equal 0.

As a result, we can calculate the derivatives of $\Phi_{\mathbf{w}^\alpha}(\widetilde{\theta}_\alpha)$ w.r.t. $\widetilde{\theta}_\alpha(x_\alpha)$ recursively as,

$$
\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial \widetilde{\theta}_\alpha(x_\alpha)} = \frac{\partial \log Z_c}{\partial \widetilde{\theta}_\alpha(x_\alpha)} = \mu_\alpha(x_c) \cdot \frac{\partial \log Z_{c-1}(x_c)}{\partial \widetilde{\theta}_\alpha(x_\alpha)} = \cdots = \prod_{i=1}^c \mu_\alpha(x_i | x_{i+1:c}) = \mu_\alpha(x_\alpha). \quad (17)
$$

By the chain rule,

$$
\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial \delta_i^\alpha(x_i)} = \sum_{x_{\alpha \setminus i}} \frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial \widetilde{\theta}_\alpha(x_i, x_{\alpha \setminus i})} \cdot \frac{\partial \widetilde{\theta}_\alpha(x_i, x_{\alpha \setminus i})}{\partial \delta_i^\alpha(x_i)} = - \sum_{x_{\alpha \setminus i}} \mu_\alpha(x_\alpha),
$$

then (14) has been proved.

Applying the variational form of powered-sum (3) to $\Phi_{\mathbf{w}^\alpha}$, we have

$$
\Phi_{\mathbf{w}^\alpha}(\widetilde{\theta}_\alpha) = \max_{b_\alpha \in \mathbb{M}_\alpha(G)} \left\{ \langle \widetilde{\theta}_\alpha, b_\alpha \rangle + \sum_i w_i^\alpha H(x_i | x_{i+1:n}; b_\alpha) \right\}.
$$

According to Danskin's theorem, the derivative $\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial \widetilde{\theta}_\alpha(x_\alpha)} = b_\alpha^*(x_\alpha)$, which is the optimum of RHS. Combined with (17), we have $b_\alpha^* = \mu_\alpha$ immediately, and the derivative w.r.t. $w_i^\alpha$ is,

$$
\frac{\partial \Phi_{\mathbf{w}^\alpha}}{\partial w_i^\alpha} = H(x_i | x_{i+1:c}; \mu_\alpha(x_\alpha)),
$$

then (15) has been proved.

$\square$

### References

[1] T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010.

[2] Q. Liu. *Reasoning and Decisions in Probabilistic Graphical Models–A Unified Framework*. PhD thesis, University of California, Irvine, 2014.

[3] Q. Liu and A. Ihler. Bounding the partition function using Hölder's inequality. In *ICML*, 2011.

[4] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *Information Theory, IEEE Transactions on*, 51(7):2313–2335, 2005.

[5] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[6] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.