Accepted to IEEE Transactions on Information Theory
This version: March 2005

A new class of upper bounds on the log partition function

M. J. Wainwright, T. S. Jaakkola and A. S. Willsky

***Abstract*— We introduce a new class of upper bounds on the log partition function of a Markov random field. This quantity plays an important role in various contexts, including approximating marginal distributions, parameter estimation, combinatorial enumeration, statistical decision theory, and large deviations bounds. Our derivation is based on concepts from convex duality and information geometry: in particular, it exploits mixtures of distributions in the exponential domain, and the Legendre mapping between exponential and mean parameters. In the special case of convex combinations of tree-structured distributions, we obtain a family of variational problems, similar to the Bethe variational problem, but distinguished by the following desirable properties: (i) they are convex, and have a unique global optimum; and (ii) the optimum gives an upper bound on the log partition function. This optimum is defined by stationary conditions very similar to those defining fixed points of the sum-product algorithm, or more generally any local optimum of the Bethe variational problem. As with sum-product fixed points, the elements of the optimizing argument can be used as approximations to the marginals of the original model. The analysis extends naturally to convex combinations of hypertree-structured distributions, thereby establishing links to Kikuchi approximations and variants.**

Keywords: Approximate inference; Belief propagation; Bethe/Kikuchi free energy; Factor graphs; Graphical models; Information Geometry; Markov random field; Partition function; Sum-product algorithm; Variational method.

I. Introduction

Undirected graphical models, otherwise known as Markov random fields, provide a powerful framework with which to represent a structured set of dependency relations among a set of random variables [e.g., 13], [24]. Such models are used in a wide variety of fields, including statistical physics, coding theory, statistical image processing, computer vision, and machine learning. Associated with any Markov random field is a *log partition function*, the most obvious role of which is to normalize the distribution. In addition to this purpose, it plays a fundamental role in various contexts, including approximate inference [17], maximum likelihood parameter estimation [19], combinatorial enumeration [34], and large deviations bounds [12]. For an undirected graphical model without cycles (i.e., a tree), the log partition function can be computed in a straightforward manner by recursive updates of a dynamic-programming nature [30]. For a general Markov random field on a graph with cycles, however, exact calculation of the log partition function is computationally intractable due to the exponential number of terms. Therefore, approximating or obtaining bounds on the log partition function is an important problem.

There is a fairly substantial literature on the use of Monte Carlo methods for approximating the log partition function in a stochastic manner [e.g., 18], [31]. It is also of considerable interest to obtain deterministic upper and lower bounds on the log partition function. In this context, mean field theory [e.g., 48], [19] is well-known to provide a lower bound on the log partition function. By using higher order expansions, Leisink and Kappen [25] derived lower bounds that are tighter than naive mean field. In contrast, upper bounds on the log partition function are not widely available. For the special case of the Ising model,[1] Jaakkola and Jordan [17] developed a recursive node-elimination procedure for upper bounding the log partition function. However, this procedure does not appear to have any straightforward generalizations to more complex Markov random fields.

In this paper, we develop a new class of upper bounds on the partition function of an arbitrary Markov random field. The basic idea is to approximate the original distribution using a collection of tractable distributions, where the term "tractable" refers to a distribution for which the partition function can be calculated efficiently by a recursive algorithm. The canonical example of a tractable Markov random field is one corresponding to a graph without any cycles (i.e., a tree). More generally, distri-

M. J. Wainwright (wainwrig@eecs.berkeley.edu) is with the Department of Electrical Engineering and Computer Science and the Department of Statistics, UC Berkeley, CA. T. S. Jaakkola (tommi@csail.mit.edu) and A. S. Willsky (willsky@mit.edu) are with the Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA. This work first appeared in the thesis [35], and subsequently in conference papers [36], [38]. Work was supported in part by ODDR&E MURI Grant DAAD19-00-1-0466 through the Army Research Office; by the Office of Naval Research through Grant N00014-00-1-0089; and by the Air Force Office of Scientific Research through Grant F49620-00-1-0362.

[1] The Ising model [4] is a pairwise Markov random field on a binary random vector.

butions corresponding to graphs of bounded treewidth are tractable, in that the junction tree algorithm [24] can be used to perform exact calculations, albeit with a cost exponential in the treewidth. Although the ideas and analysis described here can be applied quite generally to approximations based on bounded treewidth graphs, the primary focus of this paper is the use of approximations based on spanning trees.

One cornerstone of our work is provided by exponential representations of distributions, which have been studied extensively in statistics and applied probability [e.g., 1], [3], [9]. In particular, the entire collection of Markov random fields associated with a given graph constitutes an exponential family. Any member of the family is specified by an exponential parameter, the elements of which are weights for potential functions defined on the graph cliques. Given some target distribution, we decompose its exponential parameter as a convex combination of exponential parameters corresponding to tractable distributions. By exploiting the convexity of the log partition function, such a mixture in the exponential domain leads to an upper bound on the log partition function of the target distribution. The collection of weights defining the convex combination itself can be interpreted as a probability distribution over the set of tractable distributions.

Of course, even in the case of spanning trees, there is a huge collection of such bounds — one for every collection of tree exponential parameters, and corresponding distribution over these trees. It is natural, then, to consider the problem of optimizing the choice of these free variables so as to obtain the tightest possible upper bound. At first sight, solving this problem appears to be intractable. Even with the distribution over the spanning trees fixed, there are more free variables in the problem than spanning trees in the graph, the number of which is very large for a reasonably complex graph with cycles. However, by construction, the problem involves a convex cost with linear constraints, so that it is amenable to the methods of convex duality [6], [32]. In particular, by a Lagrangian dual reformulation, we obtain a variational problem that can be solved efficiently, thereby yielding the value of the upper bound that is optimal over all exponential parameters on all spanning trees of the graph.

Interestingly, the dual function obtained by this Lagrangian reformulation turns out to be closely related to the Bethe variational problem of statistical physics. Thus, our analysis makes connections with the recent work of Yedidia, Freeman and Weiss [46], who showed that the well-known belief propagation or sum-product algorithm [e.g., 23], [30] can be formulated as a method for attempting to solve this Bethe variational problem. Not surprisingly then, the conditions defining the optima of our dual problem are strikingly similar to the conditions that characterize fixed points of the sum-product algorithm. (See [37], [35] for more details of the tree-based reparameterization interpretation of sum-product and related algorithms). So as to make the connection with the sum-product algorithm even more concrete, we develop a tree-reweighted sum-product algorithm, which can be used to solve our variational problem. Despite these similarities, our dual function has two properties that are not typically enjoyed by the Bethe formulation: it is convex, and the global optimum gives an upper bound on the log partition function.

We then turn to the problem of optimizing the choice of distribution over spanning trees of the graph. Here the exponential explosion in the problem dimension again poses a challenge. However, we are able to exploit the fact that our dual function depends on the spanning tree distribution only via a set of so-called edge appearance probabilities, one for each edge of the graph. We show that this edge appearance vector must belong to the *spanning tree polytope.* This set, which is a well-studied object in combinatorial optimization and matroid theory [e.g., 14], [42], is characterized by a number of inequalities that is exponential in the graph size, thereby precluding a direct approach to optimizing our nonlinear dual function. However, maximizing a linear function over the spanning tree polytope is equivalent to solving a maximum weight spanning tree problem, which can be performed efficiently by a greedy algorithm [20]. Thus, it is feasible to optimize the edge appearance probabilities via a conditional gradient method [6]. We show that this algorithm has the conceptually appealing interpretation of sequentially fitting the current data with the spanning tree that is best in the maximum likelihood sense, as in the work of Chow and Liu [11].

The remainder of this paper is organized in the following manner. In Section II, we introduce the necessary background for subsequent development, beginning with the basics of graph theory and then turning to Markov random fields (MRFs). Our presentation of MRFs is in terms of exponential representations, thereby enabling us to harness associated results from information geometry [e.g., 1], [3]. In Section III, we develop the basic form of the upper bounds, obtained by taking convex combinations of tree-structured distributions. Section IV is devoted to analysis of the optimal form of the upper bounds, as well as discussion of their properties. In Section V, we present efficient algorithms for computing the optimal form of these upper bounds. Section VI describes the extensions of our techniques and analysis to convex combinations of hypertree-structured distributions. We conclude in Section VII with a discussion, and directions for future research.

II. Background

In this section, we first provide some basic concepts from graph theory that are necessary for development in the sequel; further background can be found in various sources [e.g., 5], [7], [8]. We then introduce the concept of a Markov random field (MRF), with particular emphasis on exponential families of MRFs. More details on graphical models and Markov random fields can be found in the books [e.g., 19], [13], [24]. Next we present a number of results from convex duality and information geometry that are associated with exponential families. Our treatment emphasizes only those concepts necessary for subsequent developments; more background on information geometry and convex duality can be found in a variety of sources [e.g., 1], [2], [3], [32], [39].

A. Basics of graph theory

An undirected graph $G = (V, E)$ consists of a set of nodes or vertices $V = \{1, \ldots, N\}$ that are joined by a set of edges E. This paper focuses exclusively on *simple* graphs, in which multiple edges between the same pair of vertices, as well as self-loops (i.e., an edge from a vertex back to itself) are forbidden. A *clique* of the graph G is any subset of the vertex set V for which each node is connected to every other. A clique is *maximal* if it is not properly contained within any other clique. For each $s \in V$, we let $\Gamma(s) = \{ \; t \in V \mid (s,t) \in E \; \}$ denote the set of *neighbors* of s.

A *path* from node s_0 to node s_k is a sequence of distinct edges $\{(s_0, s_1), (s_1, s_2), \ldots (s_{k-1}, s_k) \; \}$. We say that a graph is *connected* if for each pair $\{s, t\}$ of distinct vertices, there is a path from s to t. A *component* of a graph is a maximal connected subgraph. A *cycle* in a graph is a path from a node s back to itself; that is, a cycle consists of a sequence of distinct edges $\{ \; (s_0, s_1), \; (s_1, s_2), \; \ldots, (s_{k-1}, s_k) \; \}$ such that $s_0 = s_k$. A *tree* $T = (V(T), E(T))$ is a cycle-free graph consisting of a single connected component; a *forest* is a disjoint union of trees. A tree is *spanning* if it reaches every vertex (i.e., $V(T) = V$). See Figure 1 for illustration of these concepts. Given a graph G with a single connected component, a *vertex cutset* is any subset $B \subset V$ whose removal breaks the graph into two or more pieces. For example, with reference to the graph of Figure 2, the subset of nodes B is a vertex cutset, because it separates the graph into the disjoint parts A and C.

B. Exponential representation of Markov random fields

An undirected graph $G = (V, E)$ defines a Markov random field (MRF) in the following way. We first associate to each vertex $s \in V$ a random variable x_s taking values in some sample space $\mathcal{X}_s$. The focus of this paper is the discrete case for which $\mathcal{X}_s = \{0, \ldots, m_s - 1\}$. We let $\mathbf{x} = \{ \; x_s \mid s \in V\}$ be a random vector with $N = |V|$ elements taking values in the Cartesian product space $\mathcal{X}^N = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N$. For any subset $A \subset V$, we let $\mathbf{x}_A$ denote the collection $\{x_s \mid s \in A\}$ of random variables associated with nodes in A.

Of interest are random vectors $\mathbf{x}$ that are *Markov* with respect to the graph G. To define this Markov property, let A, B and C be arbitrary subsets of the vertex set V, and let $\mathbf{x}_{A|B}$ denote the random vector $\mathbf{x}_A$ conditioned on $\mathbf{x}_B$. The random vector $\mathbf{x}$ is Markov with respect to the graph if $\mathbf{x}_{A|B}$ and $\mathbf{x}_{C|B}$ are conditionally independent whenever B separates A and C. See Figure 2 for an illustration of this correspondence between graph separation and conditional independence.

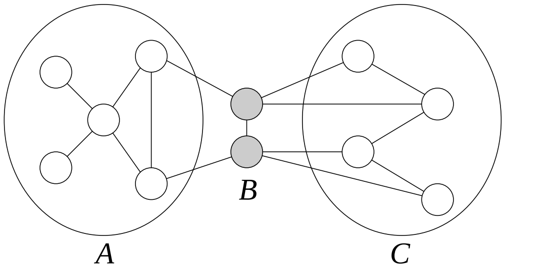


Fig. 2. Illustration of the relation between conditional independence and graph separation. Here the set of nodes B separates A and C, so that for a Markov random field, $\mathbf{x}_{A|B}$ and $\mathbf{x}_{C|B}$ must be conditionally independent.

The well-known *Hammersley-Clifford theorem* [e.g., 24] asserts that any Markov random field $p(\mathbf{x})$ that is strictly positive (i.e., $p(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^N$) decomposes in terms of functions associated with the cliques of the graph. To be more precise, a *potential function* associated with a given clique C is mapping $\phi : \mathcal{X}^N \to \mathbb{R}$ that depends only on the subcollection $\mathbf{x}_C = \{x_s \mid s \in V\}$. There may be a family of potential functions $\{\phi_\alpha \mid \alpha \in \mathcal{I}(C)\}$ associated with any given clique, where α is an index ranging over some set $\mathcal{I}(C)$. Taking the union over all cliques defines the overall index set $\mathcal{I} = \cup_C \mathcal{I}(C)$. The full collection of potential functions $\{\phi_\alpha \mid \alpha \in \mathcal{I}\}$ defines a vector-valued mapping $\boldsymbol{\phi} : \mathcal{X}^N \to \mathbb{R}^d$, where $d = |\mathcal{I}|$ is the total number of potential functions. Associated with $\boldsymbol{\phi}$ is a real-valued vector $\theta = \{ \; \theta_\alpha \mid \alpha \in \mathcal{I} \; \}$, known as the exponential parameter vector. For a fixed $\mathbf{x} \in \mathcal{X}^N$, we use $\langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle$ to denote the ordinary Euclidean product (in $\mathbb{R}^d$) between θ and $\boldsymbol{\phi}(\mathbf{x})$.

The *exponential family* associated with $\boldsymbol{\phi}$ consists of the following parameterized collection of Markov random fields:

$$p(\mathbf{x}; \theta) \quad = \quad \exp\big\{\langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle \; - \; \Phi(\theta)\big\}, \qquad \text{(1a)}$$

$$\Phi(\theta) \quad = \quad \log\big(\sum_{\mathbf{x} \in \mathcal{X}^N} \exp\{\langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle\}\big). \qquad \text{(1b)}$$

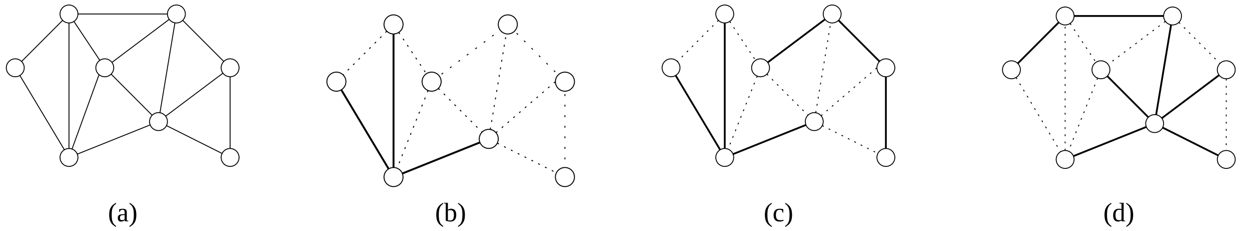


Fig. 1. (a) Graph with cycles. (b) A tree is a cycle-free subgraph. (c) A forest is a disjoint union of trees; it is spanning if it reaches every vertex of the graph. (d) A spanning tree reaches every vertex of the graph.

Each vector $\theta \in \mathbb{R}^d$ indexes a particular Markov random field $p(\mathbf{x};\theta)$ in this exponential family; with some abuse of notation, we will often use the parameter vector θ itself as a shorthand for the associated distribution. Of central interest in this paper is the quantity Φ defined in equation (1b): it is the *log partition function* that serves to normalize the distribution. Note that it is defined by a summation over all configurations $\mathbf{x} \in \mathcal{X}^N$, the number of which grows exponentially in the number of vertices N.

The *Ising model* of statistical physics [e.g., 4] provides a simple illustration of an exponential family.[2] It involves a binary random vector $\mathbf{x} \in \{0,1\}^N$, with a distribution defined by a graph with maximal cliques of size two (i.e., edges):

$$p(\mathbf{x};\theta) = \exp\big\{\sum_{s\in V}\theta_s x_s + \sum_{(s,t)\in E}\theta_{st}x_s x_t - \Phi(\theta)\big\}. \tag{2}$$

Here θ_{st} is the strength of edge (s,t), and θ_s is the node parameter for node s. In this case, the index set $\mathcal{I}$ consists of the union $V \cup E$. The exponential representation in the Ising model is *minimal* [3], because there are no linear combinations of the potentials $\boldsymbol{\phi} = \{x_s, s \in V\} \cup \{x_s x_t, (s,t) \in E\}$ equal to a constant for all $\mathbf{x} \in \{0,1\}^N$.

In contrast to a minimal representation, it is often convenient to use an *overcomplete* exponential representation, in which the potential functions $\boldsymbol{\phi}$ satisfy linear constraints. More specifically, we will use an overcomplete representation in which the basic building blocks are indicator functions of the form $\delta_{s;j}(x_s)$ — the function that is equal to one if $x_s = j$, and zero otherwise. As an illustration, for a pairwise MRF (i.e., defined on a graph with maximal cliques of size two), we use the following collection of potential functions:

$$\{\delta_{s;j}(x_s) \;\big|\; j \in \mathcal{X}_s\,\} \quad \text{for } s \;\in\; V, \tag{3a}$$
$$\{\delta_{s;j}(x_s)\delta_{t;k}(x_t) \;\big|\; (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\,\} \quad \text{for } (s,t) \;\in\; E. \tag{3b}$$

The overcompleteness of the representation is manifest in various linear constraints among the potentials (e.g., $\delta_{s;j}(x_s) - \sum_{k\in\mathcal{X}_t}\delta_{s;j}(x_s)\delta_{t;k}(x_t) = 0$). As a consequence of this overcompleteness, there are many exponential parameters corresponding to a given distribution (i.e., $p(\mathbf{x};\theta) = p(\mathbf{x};\widetilde{\theta})$ for $\theta \neq \widetilde{\theta}$). As a particular example, consider a Bernouilli random variable with a minimal exponential distribution $p(x;\gamma) = \exp\{\gamma x - \Phi(\gamma)\}$. In this very special case, the overcomplete representation of equation (3) takes the form

$$p(x;\theta) \quad = \quad \exp\{\theta_0(1-x) + \theta_1 x - \Phi(\theta)\}.$$

By inspection, any parameter $\theta = (\theta_0,\theta_1)$ that satisfies the linear constraint $\theta_1 - \theta_0 = \gamma$ gives rise to the same distribution $p(\mathbf{x};\gamma)$. Despite this many-to-one correspondence between parameters and distributions, we will make considerable use of this overcomplete representation, since it leads to more easily interpretable results.

C. *Significance of graph structure*

We now consider the complexity of computing, for a given Markov random field $p(\mathbf{x};\theta)$, the value of the log partition function $\Phi(\theta)$ in equation (1b). A brute force approach, which entails a summation over a number of terms $|\mathcal{X}^N|$ that grows exponentially in N, is not feasible for large problems. It turns out that the inherent complexity of this problem depends critically on the nature of the underlying graph. Any tree-structured graph can be "rooted" by specifying some vertex as the root. The log partition function can then be computed by a sequence of recursive computations, sweeping upwards from the leaves to the root of the tree. The overall computational complexity of such an algorithm is $\mathcal{O}(m^2 N)$, where $m = \max_{s\in V}|m_s|$. More details on such dynamic-programming algorithms for trees can be found in various sources [e.g., 30], [23], [45].

Any graph with cycles can be converted, through a process of clustering its nodes together so as to form aggregated nodes, into a structure known as a junction tree [13], [24]. At least in principle, standard tree algorithms can be applied to perform exact computations

[2] To be precise, the model presented here is slightly more general than the classical Ising model, since it allows the parameter for each node and edge to vary independently.

on this junction tree. This combination of forming a junction tree and then running a standard tree algorithm to perform exact computation is known as the *junction tree algorithm* [24]. As noted above, however, the cost of running a tree algorithm depends quadratically on the number of states at each node. This number of states grows exponentially with the size of the largest cluster involved in the junction tree, a quantity closely related to the *treewidth* of the graph.[3] For many classes of graphs, this treewidth grows sufficiently rapidly (as a function of N) so as to render prohibitive the cost of running an exact tree algorithm on the junction tree. This explosion is an explicit demonstration of the intrinsic complexity of exact computations for graphs with cycles.

Throughout this paper, we use the word *tractable* to refer to either a tree-structured distribution (or more generally, a distribution associated with a graph of bounded treewidth) for which it is computationally feasible to apply the junction tree algorithm [24]. Of interest to us, of course, is the approximate computation of the log partition function for an *intractable* model.

D. Information geometry and convex duality

In this section, we provide a brief overview of the ideas from convex duality and information geometry that are necessary for development in the sequel. It is well-known [e.g., 2] that the log partition function Φ is convex as a function of the exponential parameters. This convexity follows from properties of Φ given in the following lemma:

Lemma 1. (a) *Taking first derivatives of Φ generates (first-order) moments of $\boldsymbol{\phi}$ — viz.:*

$$\frac{\partial \Phi(\theta)}{\partial \theta_\alpha} \;=\; \mathbb{E}_\theta[\phi_\alpha(\mathbf{x})] \;=\; \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x};\theta)\phi_\alpha(\mathbf{x}).$$

(b) *Moreover, the second derivatives $\frac{\partial^2 \Phi(\theta)}{\partial \theta_\alpha \, \partial \theta_\beta}$ are covariance terms $\operatorname{cov}_\theta\{\phi_\alpha, \, \phi_\beta\}$ of the form*

$$\mathbb{E}_\theta[\phi_\alpha(\mathbf{x})\phi_\beta(\mathbf{x})] - \mathbb{E}_\theta[\phi_\alpha(\mathbf{x})]\mathbb{E}_\theta[\phi_\beta(\mathbf{x})].$$

More generally, the log partition function Φ is the cumulant generating function, in that the partial derivatives of order n correspond to n^{th}-order cumulants of $\boldsymbol{\phi}$. Note that Lemma 1(b) implies that Φ is convex, since any covariance matrix must be positive semidefinite. Moreover, the convexity is strict for a minimal representation.

An important quantity is the conjugate dual function of Φ. It is defined by the optimization problem

$$\Phi^*(\mu) = \sup_{\theta \in \mathbb{R}^d} \{\langle \theta, \, \mu \rangle - \Phi(\theta)\}, \tag{4}$$

where $\mu \in \mathbb{R}^d$ is a vector of dual variables. For a given dual vector μ^*, it can be shown [39] that the supremum in equation (4) is either equal to $+\infty$, or is attained at a vector θ^* such that the following condition holds for each $\alpha \in \mathcal{I}$:

$$\mu^*_\alpha \;=\; \mathbb{E}_{\theta^*}[\phi_\alpha(\mathbf{x})] \;:=\; \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x};\theta^*)\phi_\alpha(\mathbf{x}). \tag{5}$$

These moment-matching conditions can be obtained by using Lemma 1 to calculate the gradient of $\langle \theta, \, \mu \rangle - \Phi(\theta)$ with respect to θ, and then setting this gradient to zero. Since equation (5) involves taking an expectation, the dual variables are often referred to as *mean or moment parameters*. Note that in order for equation (5) to have a solution, the dual vector μ^* must arise as a vector of *realizable* mean parameters; more precisely, it can be shown [39] that it must belong to the (relative interior) of the set

$$\operatorname{MARG}(\boldsymbol{\phi}) = \big\{ \; \mu \in \mathbb{R}^d \; \big| \; \exists \;\; p(\cdot) \;\; \text{s.t.} \;\; \mathbb{E}_p[\phi(\mathbf{x})] = \mu \big\}. \tag{6}$$

More details on the structure of this so-called marginal polytope can be found in the technical report [39].

In order to calculate an explicit form for the Legendre dual Φ^*, we substitute the result of equation (5) into equation (4), which leads to:

$$\begin{aligned} \Phi^*(\mu^*) \;&=\; \langle \mu^*, \, \theta^* \rangle - \Phi(\theta^*) \\ &=\; \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x};\theta^*) \log p(\mathbf{x};\theta^*). \end{aligned} \tag{7}$$

Therefore, whenever $\mu^* \in \operatorname{MARG}(\boldsymbol{\phi})$, the value of the conjugate dual $\Phi^*(\mu^*)$ is equal to the negative entropy of the distribution $p(\mathbf{x};\theta^*)$.

The conjugate duality between Φ and Φ^* defines a mapping $\Lambda \, : \, \theta \mapsto \mu$ between exponential and mean parameters. For a minimal representation, the function Φ is strictly convex, and the function is Λ is one-to-one and hence invertible on the its image [see 9], [39], which is the relative interior of $\operatorname{MARG}(\boldsymbol{\phi})$. On the basis of these mappings, we can specify distributions either in terms of the exponential parameter θ, or the associated dual parameter μ. Given a valid dual parameter μ, the notation $p(\mathbf{x};\mu)$ will be used as a shorthand for the equivalent exponential distribution $p(\mathbf{x};\Lambda^{-1}(\mu))$. Any pair θ and μ that are related via equation (7) are said to be *dually coupled*.

III. Upper bounds by convex combinations

In this section, we develop the basic form of the upper bounds on the log partition function Φ. For simplicity in exposition, the bulk of our development focuses on the case of a so-called pairwise Markov random field, for which the maximal cliques of the underlying graph have size two. In this case, the collection $\boldsymbol{\phi}$ of potentials

[3] To be more precise, the treewidth k of a graph G is equal to $c-1$, where c is the size of the largest cluster in a minimal junction tree of G.

consists only of functions associated with single nodes and edges (i.e., pairs of nodes). However, as we show in Section VI, the analysis given here is applicable also to general Markov random fields that may include higher order cliques.

In the case of pairwise MRFs, our upper bounds are based on convex combinations of tree-structured distributions; accordingly, we begin with the notation and definitions necessary to specify such combinations. Our results are most clearly stated in the overcomplete exponential representation of equation (3), which is based on indicator functions at single nodes and on edges. In this case, the index set $\mathcal{I}$ is given by the union

$$\{(s;j) \mid s \in V,\ j \in \mathcal{X}_s\} \ \cup \\ \{(st;jk) \mid (s,t) \in E,\ (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}.$$

Let $\bar{\theta} = \{\bar{\theta}_\alpha \mid \alpha \in \mathcal{I}\}$ be the exponential parameter corresponding to the *target distribution* $p(\mathbf{x};\bar{\theta})$ of interest. Note that $\bar{\theta}$ is a vector in $\mathbb{R}^d$, where $d = |\mathcal{I}| = \sum_{s\in V} m_s + \sum_{(s,t)\in E} m_s m_t$.

In order to obtain bounds, we will consider only *spanning trees* of the graph.[4] Accordingly, let $\mathfrak{T} = \mathfrak{T}(G)$ denote the set of all spanning trees of the graph G. For each spanning tree $T \in \mathfrak{T}$, let $\theta(T) \in \mathbb{R}^d$ be an exponential parameter vector that respects the structure of the tree T. To be explicit, suppose that $T = V, E(T)$, and let $\mathcal{I}(T) \subset \mathcal{I}$ be the index subset formed by indices associated with single nodes, or with edges in the tree:

$$\mathcal{I}(T) = \{(s;j) \mid s \in V,\ j \in \mathcal{X}_s\} \ \cup \\ \{(st;jk) \mid (s,t) \in E(T),\ (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}.$$

In order for the distribution $p(\mathbf{x};\theta(T))$ to be tree-structured, the parameter $\theta(T)$ must belong to the following constraint set:[5]

$$\mathcal{E}(T) \quad = \quad \{\ \theta(T) \in \mathbb{R}^d \mid \theta_\alpha(T) = 0 \ \ \forall \ \alpha \in \mathcal{I}\backslash\mathcal{I}(T)\}. \tag{8}$$

For compactness in notation, let $\boldsymbol{\theta} := \{\theta(T) \mid T \in \mathfrak{T}\}$ denote the full collection of tree-structured exponential parameter vectors, where $\theta(T)$ indexes those subelements of $\boldsymbol{\theta}$ associated with spanning tree T. The full collection $\boldsymbol{\theta}$ is required to belong the affine constraint set

$$\mathcal{E} \quad := \quad \{\boldsymbol{\theta} \in \mathbb{R}^{d\times|\mathfrak{T}(G)|} \mid \theta(T) \in \mathcal{E}(T) \ \text{ for all } \ T \ \in \ \mathfrak{T}(G)\}. \tag{9}$$

In order to define a convex combination, we require a probability distribution $\vec{\boldsymbol{\rho}}$ over the set of spanning trees

$$\vec{\boldsymbol{\rho}} \quad := \quad \{\ \rho(T),\ T \in \mathfrak{T} \mid \rho(T) \geq 0, \quad \sum_{T\in\mathfrak{T}} \rho(T) = 1\}. \tag{10}$$

For any distribution $\vec{\boldsymbol{\rho}}$, we define its *support* to be the set of trees to which it assigns strictly positive probability; that is

$$\text{supp}(\vec{\boldsymbol{\rho}}) \quad := \quad \{\ T \in \mathfrak{T} \mid \rho(T) > 0\ \}. \tag{11}$$

For a given tree $T \in \mathfrak{T}$, let $\nu(T) \in \{0,1\}^{|E|}$ be an indicator vector for the edges that comprise the tree, with element (s,t) give by

$$\big[\nu(T)\big]_{st} \quad := \quad \begin{cases} 1 & \text{if } (s,t) \in T \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The *spanning tree polytope* [14], [10], which we denote by $\mathbb{T}(G)$, is defined as the convex hull of these tree indicator vectors:

$$\mathbb{T}(G) := \{\boldsymbol{\rho_e} \in \mathbb{R}^{|E|} \bigm| \exists\ \vec{\boldsymbol{\rho}} \ \text{s.t.}\ \boldsymbol{\rho_e} = \sum_{T\in\mathfrak{T}} \rho(T)\nu(T)\}, \tag{13}$$

where $\vec{\boldsymbol{\rho}}$ ranges over all possible distributions over spanning trees. Note that for a fixed edge $(s,t) \in E$, the element $\rho_{st} = [\boldsymbol{\rho_e}]_{st}$ can be interpreted as the probability $\Pr_{\vec{\boldsymbol{\rho}}}\{(s,t) \in T\}$ that edge $(s,t) \in E$ appears in a spanning tree T chosen randomly under $\vec{\boldsymbol{\rho}}$. Thus, the vector $\boldsymbol{\rho_e} = \{\rho_{st} \mid (s,t) \in E\}$ corresponds to the full collection of these *edge appearance probabilities*, and we refer to it as an *edge appearance vector*. See Figure 3 for an illustration of the edge appearance vector $\boldsymbol{\rho_e}$ and the spanning tree polytope. Throughout this paper, we assume that $\vec{\boldsymbol{\rho}}$ is chosen such that the associated edge appearance probabilities $\rho_e = \Pr_{\vec{\boldsymbol{\rho}}}\{e \in T\}$ are all strictly positive; that is, each edge $e \in E$ appears in at least one tree $T \in \text{supp}(\vec{\boldsymbol{\rho}})$. We say that the distribution $\vec{\boldsymbol{\rho}}$, or the edge appearance probabilities $\boldsymbol{\rho_e}$ are *valid* when they satisfy this condition.

A *convex combination* of exponential parameter vectors is given by taking an expectation with respect to $\vec{\boldsymbol{\rho}}$ as follows:

$$\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] \quad := \quad \sum_{T\in\mathfrak{T}} \rho(T)\theta(T). \tag{14}$$

This combination represents a mixture of distributions in the exponential domain [2], [1]. With reference to the target distribution $p(\mathbf{x};\bar{\theta})$, we are especially interested in collections of exponential parameters $\boldsymbol{\theta}$ for which there exists a convex combination that is equal to $\bar{\theta}$. Accordingly, we define the following set:

$$\mathcal{A}(\bar{\theta}) \quad := \quad \big\{(\boldsymbol{\theta};\vec{\boldsymbol{\rho}}) \bigm| \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] = \bar{\theta}\big\}. \tag{15}$$

It is not difficult to see that $\mathcal{A}(\bar{\theta})$ is never empty.

Example 1 (Single cycle graph). As an illustration of these definitions, consider a binary distribution defined by a single cycle on 4 nodes. We define the target distribution in the Ising form

$$p(\mathbf{x};\bar{\theta}) \quad = \quad \exp\{x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 - \Phi(\bar{\theta})\}.$$

[4] Our methods could be applied more generally to spanning forests, but they would lead to weaker bounds.

[5] Since the constraints on $\theta(T)$ are affine, each such $\mathcal{E}(T)$ is an e-flat manifold in the sense of information geometry [2], [1].

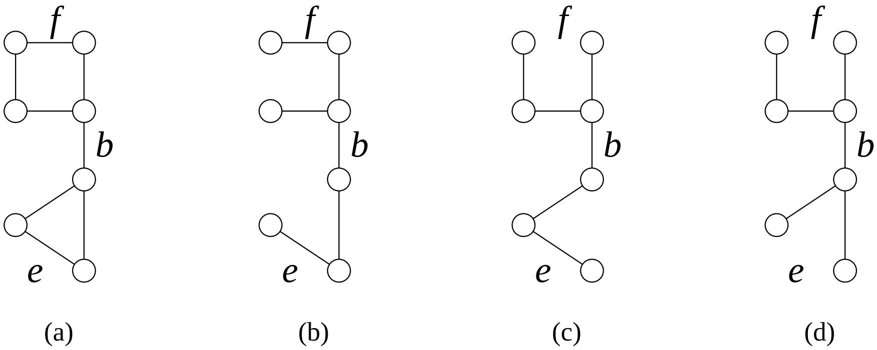


Fig. 3. Illustration of the spanning tree polytope $\mathbb{T}(G)$. Original graph is shown in panel (a). Probability $1/3$ is assigned to each of the three spanning trees $\{\, T_i \mid i = 1, 2, 3\,\}$ shown in panels (b)–(d). Edge b is a so-called bridge in G, meaning that it must appear in any spanning tree. Therefore, it has edge appearance probability $\rho_b = 1$. Edges e and f appear in two and one of the spanning trees respectively, which gives rise to edge appearance probabilities $\rho_e = 2/3$ and $\rho_f = 1/3$.

That is, the target distribution is specified by the minimal parameter $\bar{\theta} = [0\ 0\ 0\ 0 \quad 1\ 1\ 1\ 1]$, where the zeros represent the fact that $\bar{\theta}_s = 0$ for all $s \in V$. The four possible spanning trees $\mathfrak{T} = \{T_i \mid i = 1, \ldots, 4\,\}$ of the single cycle with four nodes are illustrated in Figure 4. We define a set of associated exponential parameters $\boldsymbol{\theta} = \{\theta(T_i)\}$ as follows:

$$\begin{array}{rcl} \theta(T_1) & = & (4/3)\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 1 & 1 & 0\end{bmatrix} \\ \theta(T_2) & = & (4/3)\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 1 & 0 & 1\end{bmatrix} \\ \theta(T_3) & = & (4/3)\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix} \\ \theta(T_4) & = & (4/3)\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 1 & 1\end{bmatrix}. \end{array}$$

Finally, we choose $\rho(T_i) = 1/4$ for all $T_i \in \mathfrak{T}$. With this uniform distribution over trees, we have $\rho_e = 3/4$ for each edge, and moreover, $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] = \bar{\theta}$ so that the pair $(\boldsymbol{\theta}; \vec{\boldsymbol{\rho}})$ belongs to $\mathcal{A}(\bar{\theta})$. □

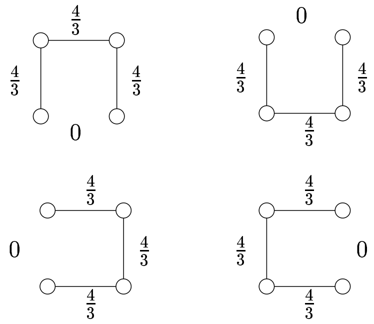


Fig. 4. A convex combination of four distributions $p(\mathbf{x}; \theta(T_i))$, each defined by a spanning tree T_i, is used to approximate the target distribution $p(\mathbf{x}; \bar{\theta})$ on the single-cycle graph.

The convexity of Φ allows us to apply Jensen's inequality [12] to a convex combination specified by a pair $(\boldsymbol{\theta}, \vec{\boldsymbol{\rho}}) \in \mathcal{A}(\bar{\theta})$, thereby yielding the upper bound:

$$\begin{array}{rcl} \Phi(\bar{\theta}) & = & \Phi(\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))] \\ & \leq & \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))] \; := \; \displaystyle\sum_{T \in \mathfrak{T}} \rho(T)\Phi(\theta(T)). \end{array} \qquad (16)$$

Note that the bound of equation (16) is a function of both the distribution $\vec{\boldsymbol{\rho}}$ over spanning trees, as well as the collection $\boldsymbol{\theta}$ of tree-structured exponential parameter vectors. Our goal is to optimize both of these choices so as to minimize the RHS of equation (16), thereby obtaining the tightest possible upper bound. A major challenge to be confronted is the dimensionality of the problem: the length of $\vec{\boldsymbol{\rho}}$ corresponds to the number of spanning trees in the graph, which (for many graphs) is very large. For example, the complete graph K_N on N nodes has N^{N-2} spanning trees; more generally, the number of spanning trees can be calculated via the Matrix-Tree theorem [7].

IV. Optimal forms of upper bounds

In order to obtain optimal upper bounds of the form in equation (16), we begin by fixing the probability distribution $\vec{\boldsymbol{\rho}}$ over trees, and then optimizing the choice of the collection $\boldsymbol{\theta}$ of tree-structured exponential parameters. As we demonstrate, despite the combinatorial explosion in the number of spanning trees (and hence the dimension of $\boldsymbol{\theta}$), this problem can be solved efficiently via its Lagrangian dual. Moreover, the dual formulation also sheds light on how to optimize over the spanning tree distribution $\vec{\boldsymbol{\rho}}$.

A. Optimizing with $\vec{\boldsymbol{\rho}}$ fixed

For a fixed distribution $\vec{\boldsymbol{\rho}}$, consider the constrained optimization problem:

$$\begin{cases}\min_{\boldsymbol{\theta} \in \mathcal{E}} \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))] \\ \text{such that} \;\; \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] = \bar{\theta}, \end{cases} \tag{17}$$

where $\mathcal{E}$ is defined in equation (9). Note that $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))]$ is convex as a function of the full collection

$$\boldsymbol{\theta} := \{\, \theta(T) \mid T \in \mathfrak{T} \},$$

and moreover, the associated constraint set $\{\boldsymbol{\theta} \in \mathcal{E} \mid \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] = \bar{\theta}\}$ is linear in $\boldsymbol{\theta}$. Whenever the distribution $\vec{\boldsymbol{\rho}}$ is valid (i.e., assigns strictly positive probability to each edge), then the constraint set is nonempty. Therefore, the global minimum of problem (17) could be found, at least in principle, by a variety of standard methods in nonlinear programming [6]. However, an obvious concern is the dimension of the parameter vector $\boldsymbol{\theta}$: it is directly proportional to $|\mathfrak{T}|$, the number of spanning trees in G, which (as noted above) can be very large.

1) Lagrangian duality: Fortunately, convex duality allows us to avoid this combinatorial explosion. In particular, the Lagrangian of problem (17) gives rise to a set of dual variables, which we show can be interpreted *pseudomarginals* on the nodes and edges of the graph. Remarkably, it turns out that this single collection of pseudomarginals is sufficient to specify the optimal form $p(\mathbf{x}; \theta^*(T))$ of tree-structured distribution for every spanning tree $T \in \text{supp}(\vec{\boldsymbol{\rho}})$.

To be more precise, for each node $s \in V$, let $\tau_s = \{\tau_{s;j} \mid j \in \mathcal{X}_s\}$ be a pseudomarginal vector with m_s elements; similarly, for each edge $(s,t) \in E$, let $\tau_{st} = \{\tau_{st;jk} \mid (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}$ be a pseudomarginal vector with $m_s \times m_t$ elements. On occasion, we will also use the notation $\tau_s(x_s)$ to refer to the function that takes the value $\tau_{s;j}$ when $x_s = j$; the joint function $\tau_{st}(x_s, x_t)$ is defined similarly. We use τ to denote the full collection of pseudomarginals

$$\tau = \{\tau_s, \; s \in V\,\} \,\cup\, \{\, \tau_{st}, \; (s,t) \in E\,\}. \tag{18}$$

Note that $\tau \in \mathbb{R}^d$ is a vector of the same length as $\bar{\theta}$. This vector of pseudomarginals is required to satisfy a set of local normalization and marginalization constraints; in particular, we require that they are elements of the set $\text{LOCAL}(G)$ given by the linear constraints

$$\big\{\tau \geq 0 \;\big|\; \sum_{k \in \mathcal{X}_t} \tau_{st;jk} = \tau_{s;j}, \quad \sum_{j \in \mathcal{X}_s} \tau_{s;j} = 1 \,\big\}. \tag{19}$$

A key property of this definition is that if G is a tree, then $\text{LOCAL}(G)$ is a complete description of the set of valid single node and pairwise marginal distributions. This statement follows as a special case of the junction tree theorem [13], [24].

Let $\boldsymbol{\theta}^* = \{\, \theta^*(T) \mid T \in \mathfrak{T} \,\}$ denote an optimum[6] of problem (17). The significance of τ is in specifying any such optimum in a very compact fashion. For each tree $T \in \mathfrak{T}$, let $\Pi^T(\tau)$ denote the projection of τ onto the spanning tree T. Explicitly,

$$\Pi^T(\tau) := \{\tau_s, \; s \in V\,\} \,\cup\, \{\, \tau_{st}, \; (s,t) \in E(T)\,\} \tag{20}$$

consists only of those elements of τ corresponding to single nodes, or belonging to the edge set $E(T) \subset E$ of the tree T. Any such vector $\Pi^T(\tau)$ provides an explicit construction of a tree-structured distribution $p(\mathbf{x}; \Pi^T(\tau))$ via the usual factorization of tree-structured distributions implied by the junction tree representation [24] — viz.:

$$p(\mathbf{x}; \Pi^T(\tau)) \;:=\; \prod_{s \in V} \tau_s(x_s) \quad \prod_{(s,t) \in E(T)} \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\; \tau_t(x_t)} \tag{21}$$

We will prove that the optimal dual solution τ^* specifies the full collection of optimal exponential parameters $\boldsymbol{\theta}^*$ via the relation:

$$p(\mathbf{x}; \theta^*(T)) = p(\mathbf{x}; \Pi^T(\tau^*)) \qquad \text{for all} \;\; T \in \mathfrak{T}. \tag{22}$$

Equation (22) is an explicit statement of the fact that for each tree T, the exponential parameter $\theta^*(T)$ and the mean parameter $\Pi^T(\tau^*)$ are dually coupled (see Section II-D).

The significance of equation (22) is that one set of dual parameters τ^* are shared across *all* spanning trees of the graph. As a consequence, a single collection of pseudomarginals τ^* on nodes and edges suffices to specify the full collection $\boldsymbol{\theta}^* = \{\, \theta^*(T) \mid T \in \mathfrak{T} \,\}$ of tree parameters. Consequently, the dual formulation reduces the problem dimension from the size of $\boldsymbol{\theta}$, which is proportional to $|\mathfrak{T}|$, down to the number of elements in τ^* — namely, $d = \mathcal{O}(mN + m^2|E|)$ where $m = \max_{s \in V} m_s$. It is this massive reduction in the problem dimension that permits efficient computation of the optimum. The conditions defining the optimal τ^*, given in equation (22), are very similar to the tree-based reparameterization conditions [see 37], [35] that characterize fixed points of the sum-product algorithm, and more generally local optima of the Bethe variational problem. Not surprisingly then, the dual formulation of Theorem 1 has a very close relation with this Bethe problem.

[6] Given its convexity, problem (17) has a unique minimum that is global; however, in the overcomplete parameterization of equation (3), this optimum will be attained at many points. However, our analysis shows that this is not a concern, since any of these optima are characterized by the dual coupling in equation (22).

2) Optimal upper bounds: With this intuition, we are ready to state and prove the main result of this section. Let $\vec{\boldsymbol{\rho}}$ be a valid distribution over spanning trees, and let $\boldsymbol{\rho_e} \in \mathbb{T}(G)$ be the associated vector of edge appearance probabilities. For each $s \in V$ and pseudomarginal τ_s, we define the single node entropy:

$$H_s(\tau_s) \quad = \quad -\sum_{j \in \mathcal{X}_s} \tau_{s;j} \log \tau_{s;j}. \tag{23}$$

Similarly, for each $(s,t) \in E$, we define the mutual information between x_s and x_t as measured under the joint pseudomarginal τ_{st}:

$$I_{st}(\tau_{st}) := \sum_{(j,k)} \tau_{st;jk} \log \frac{\tau_{st;jk}}{(\sum\limits_{k \in \mathcal{X}_t} \tau_{st;jk})(\sum\limits_{j \in \mathcal{X}_s} \tau_{st;jk})}. \tag{24}$$

From these building blocks, we define the following function:

$$Q(\tau; \boldsymbol{\rho_e}) \quad := \quad -\sum_{s \in V} H_s(\tau_s) + \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}). \tag{25}$$

We also write the inner product between τ and $\bar{\theta}$ in $\mathbb{R}^d$ as follows:

$$\langle \tau, \, \bar{\theta} \rangle \; := \; \sum_{\alpha \in \mathcal{I}} \tau_\alpha \bar{\theta}_\alpha \; = \\ \sum_{s \in V} \sum_{j \in \mathcal{X}_s} \tau_{s;j} \bar{\theta}_{s;j} + \sum_{(s,t) \in E} \; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \tau_{st;jk} \bar{\theta}_{st;jk}. \tag{26}$$

Theorem 1 (Optimal upper bounds). *For each fixed $\vec{\boldsymbol{\rho}} \in \mathbb{T}(G)$, the value of the best upper bound of the form* (17) *can be found by solving the following variational problem:*

$$\Phi(\bar{\theta}) \quad \leq \quad \max_{\tau \in \text{LOCAL}(G)} \left\{ \langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e}) \right\}. \tag{27}$$

For any valid $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, the function Q is strictly convex over $\text{LOCAL}(G)$, *so that the optimum on the RHS of equation* (27) *is attained at a unique vector in* $\text{LOCAL}(G)$.

Proof: We establish that the Lagrangian dual of problem (17) is given by $\langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})$. Let $\tau^* = \{\tau^*_s, \tau^*_{st}\}$ correspond to a collection of Lagrange multipliers. In particular, for each $s \in V$ and $j \in \mathcal{X}_s$, the quantity $\tau^*_{s;j}$ is associated with the constraint that $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta_{s;j}(T)] = \bar{\theta}_{s;j}$, whereas for each $(s,t) \in E$ and $(j,k) \in \mathcal{X}_s \times \mathcal{X}_t$, the quantity $\tau^*_{st;jk}$ is associated with the constraint that $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta_{st;jk}(T)] = \bar{\theta}_{st;jk}$. To be clear, we are *not* assuming that these Lagrange multipliers correspond to pseudomarginals; nonetheless, our choice of notation is deliberately suggestive, in that our proof shows that the Lagrange multipliers can be interpreted as tree-consistent pseudomarginals as in equation (18). With this notation, we form the Lagrangian $\mathcal{L}(\boldsymbol{\theta}; \tau^*; \vec{\boldsymbol{\rho}}; \bar{\theta})$

$$\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))] + \langle \tau^*, \, \bar{\theta} - \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\theta(T)] \rangle \;= \\ \langle \tau^*, \, \bar{\theta} \rangle + \mathbb{E}_{\vec{\boldsymbol{\rho}}} \big\{ \Phi(\theta(T)) - \langle \tau^*, \, \theta(T) \rangle \big\}. \tag{28}$$

In addition to the constraints that we have enforced via Lagrange multipliers, each $\theta(T)$ is restricted to belong to the affine space $\mathcal{E}(T)$ of tree-structured exponential parameters, as defined in equation (8). We enforce these constraints explicitly without Lagrange multipliers. As before, for a given tree T, we use $\mathcal{I}(T)$ to denote the subset of indices corresponding to exponential parameters that are free to vary.

Now the Lagrangian is also convex function of $\boldsymbol{\theta}$, so that it has no local minima. By taking derivatives of the Lagrangian with respect to θ_α for $\alpha \in \mathcal{I}(T)$ and using Lemma 1, we obtain the stationary conditions $\rho(T) \big\{ \mathbb{E}_{\theta^*(T)}[\phi_\alpha] - \tau^*_\alpha \big\} = 0$ for an optimum $\boldsymbol{\theta}^* = \{\theta^*(T)\}$. If $\rho(T) = 0$, then the tree parameter $\theta(T)$ plays no role in the problem, so that we can simply ignore it. Otherwise, if $\rho(T) > 0$, the Lagrange multiplier vector τ^* is connected to the optimal tree parameters $\theta^*(T)$ via the relation:

$$\mathbb{E}_{\theta^*(T)}[\phi_\alpha(\mathbf{x})] = \tau^*_\alpha \qquad \text{for all } \alpha \in \mathcal{I}(T). \tag{29}$$

Recall that the potentials ϕ_α in our overcomplete representation correspond to indicator functions (see equation (3)). Consequently, the expectations $\mathbb{E}_{\theta^*(T)}[\phi_\alpha(\mathbf{x})]$ correspond to elements of the marginal probabilities (for example, $\mathbb{E}_{\theta^*(T)}[\delta_{s;j}(x_s)] = p(x_s = j; \theta^*(T))$).

Therefore, equation (29) has two important implications:

(a) for all trees T and for all $s \in V$, the single node marginals $p(x_s = j; \theta^*(T))$ are all equal to a common quantity $\tau^*_{s;j}$.

(b) for all trees T for which $(s,t) \in E(T)$, the joint marginal probability $p\big((x_s, x_t) = (j,k); \theta^*(T)\big)$ is equal to $\tau^*_{st;jk}$.

In other words, for each tree T, the tree-structured exponential parameter $\theta^*(T)$ is dually coupled to corresponding set $\Pi^T(\tau^*)$ of mean parameters for that tree. The key here is that the dual coupling with τ^* holds simultaneously for all trees T of the graph.

By the conjugate duality between Φ and Φ^* given in equation (7), the following relation holds for the dually coupled pair $\Pi^T(\tau^*)$ and $\theta^*(T)$:

$$\Phi^*(\Pi^T(\tau^*)) \quad = \quad \langle \theta^*(T), \, \tau^* \rangle - \Phi(\theta^*(T)). \tag{30}$$

Recall that $\Phi^*(\Pi^T(\tau^*))$ denotes the negative entropy of the tree-structured distribution $p(\mathbf{x}; \Pi^T(\tau))$. Substituting equation (30) into the definition of the Lagrangian yields that the dual function is given by $\langle \tau, \, \bar{\theta} \rangle -$

$\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\tau))]$, where we have suppressed the "$*$" on the τ vector for notational simplicity.

From the tree factorization of equation (21), we observe that the negative entropy of any tree-structured distribution can be decomposed as follows:

$$\Phi^*(\Pi^T(\tau)) = -\sum_{s \in V} H_s(\tau_s) + \sum_{(s,t) \in E(T)} I_{st}(\tau_{st}). \quad (31)$$

This decomposition of the entropy allows us to expand the expectation $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\tau))]$ with respect to $\vec{\boldsymbol{\rho}}$ as follows:

$$\sum_T \rho(T) \Big[-\sum_{s \in V} H_s(\tau_s) + \sum_{(s,t) \in E(T)} I_{st}(\tau_{st}) \Big] = \\ -\sum_{s \in V} H_s(\tau_s) + \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}).$$

In moving from the first to second lines, we have used the fact that the trees are all spanning (so that the weights associated with each node sum to one), and the definition $\rho_{st} = \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\delta\{(s,t) \in E(T)\}]$ of the edge appearance probabilities. Thus, we have recovered the function Q given in equation (25). Each of the negative entropies Φ^* is strictly convex; since Q is a convex combination of negative entropies, it is certainly convex, and strictly so as long as $\rho_{st} > 0$ for all edges (s,t).

For each tree T, the subcollection $\Pi^T(\tau)$ must be a valid set of marginal distributions for the single nodes and edges in $E(T)$. Therefore, they must normalize suitably (i.e., $\sum_j \tau_{s;j} = 1$ for all $s \in V$). Moreover, since each edge belongs to at least one tree, we must have the marginalization constraint $\sum_j \tau_{st;jk} = \tau_{t;k}$ for each edge $(s,t) \in E$. As a consequence, the domain of the dual variables τ is precisely the constraint set $\text{LOCAL}(G)$ defined in equation (19). Since the cost function is convex and Slater's condition is satisfied, strong duality holds [6]; therefore, the optimum dual value $\max_{\tau \in \text{LOCAL}(G)} \{\langle \tau, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})\}$ is equivalent to the optimal value of the primal problem (17).

See Appendix A for an additional argument to establish that the upper bound of equation (27) holds not just for edge appearance vectors in the spanning tree polytope that satisfy $\boldsymbol{\rho_e} > 0$, but more generally for any $\boldsymbol{\rho_e} \geq 0$ in the spanning tree polytope. □

We illustrate Theorem 1 with a simple example.

Example 2. Consider a single cycle graph G on four nodes. Using the overcomplete representation (3), we form a distribution $p(\mathbf{x}; \bar{\theta})$ over a binary random vector $\mathbf{x} \in \{0,1\}^4$ with the following parameters:

$$\begin{aligned} \bar{\theta}_s &:= \begin{bmatrix} 0 & 0 \end{bmatrix}' && \text{for all } s \in V, \\ \bar{\theta}_{st} &:= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} && \text{for all } (s,t) \in E. \end{aligned}$$

The four spanning trees of this single cycle graph are illustrated in Figure 4. If we place probability $1/4$ on each of them, the corresponding edge appearance probabilities are uniform $\rho_e = (3/4)$ for all edges $e \in E$. Computing the optimal τ^* for the bound in Theorem 1, we obtain:

$$\begin{aligned} \tau_s^* &\approx \begin{bmatrix} 0.18 & 0.82 \end{bmatrix}' && \text{for all } s \in V, \\ \tau_{st}^* &\approx \begin{bmatrix} 0.07 & 0.11 \\ 0.11 & 0.71 \end{bmatrix} && \text{for all } (s,t) \in E. \end{aligned}$$

This set of mean parameters yields a value for the optimal upper bound in equation (27) of $[\langle \tau^*, \bar{\theta} \rangle - Q(\tau^*; 3/4)] \approx 4.642$, as compared to the exact log partition function $\Phi(\bar{\theta}) \approx 4.625$.

□

Remarks:

(a) In our development, we restricted attention to spanning trees of the *original graph*, as opposed to all possible spanning trees of the fully connected graph on $N = |V|$ nodes. For instance, with reference to the preceding example, we could consider all $4^2 = 16$ spanning trees on the complete graph K_4, as opposed to the four spanning trees on the single cycles. A natural question, as raised by one referee, is whether or not relaxing this restriction would lead to tighter bounds. To see that it will not improve the bounds, note that we can embed any problem on any graph G into the complete graph by augmenting the exponential parameter vector θ with zeroes in entries that correspond to edges not in G (i.e., setting $\theta_{st} = 0$ where θ_{st} is the subvector corresponding to elements associated with edge (s,t) not in G). Now consider the optimization problem in equation (27): for any choice of the singleton pseudomarginals τ_s and τ_t, the optimal choice of the joint pseudomarginal τ_{st} depends only on the negative mutual information $-\rho_{st} I_{st}(\tau) \leq 0$ (because $\theta_{st} = 0$). Therefore, regardless of the choice of singleton pseudomarginals, the maximum will entail setting τ_{st} such that $-\rho_{st} I_{st}(\tau) = 0$, and hence this edge will have no influence on the bound.

(b) As the statement of Theorem 1 is in terms of an exponential family representation, it is not immediately applicable to a graphical model which has deterministic compatibility functions (e.g., error-control codes). One way in which the theorem can be extended to such cases is by using a modified exponential family, in which the deterministic compatibility functions are absorbed into the base measure [see, e.g., 40]. Another possibility is to modify the optimization problem by setting to zero each pseudomarginal that corresponds to a zero location in the deterministic compatibility function. In this case, the tree-reweighted sum-product algorithm described in Appendix C needs to be modified accordingly.

3) Link to Bethe variational problem: Yedidia et al. [46] showed that the sum-product algorithm can be understood as attempting to solve a variational problem involving the so-called Bethe free energy from statistical physics. In other work [37], we have shown that any fixed point of the sum-product algorithm (or more generally, any local optimum of the Bethe variational problem) constitutes a tree-consistent reparameterization of the original distribution. In fact, the conditions in equation (22) that define the dual coupling between the exponential parameter vectors $\boldsymbol{\theta}^* = \{\theta^*(T)\}$ and the pseudomarginals τ^* are very similar to the conditions defining fixed points of the sum-product algorithm. On this basis, it should be expected that the problem in equation (27) is closely related to the Bethe variational problem.

At the heart of the Bethe free energy lies the following *Bethe entropy approximation*:

$$\widetilde{H}_{Bethe}(\tau) \;\; = \;\; \sum_{s\in V}(1-d_s)H_s(\tau_s) + \sum_{(s,t)\in E} H_{st}(\tau_{st}) \qquad (32)$$

In this equation, $d_s = |\Gamma(s)|$ denotes the number of neighbors of node s, whereas H_s and H_{st} correspond to the single node and joint entropies defined by the pseudomarginals τ_s and τ_{st} respectively (see equation (23)). By comparison, Theorem 1 is based on the function Q of equation (25). Consider Q in the special case $\rho_e = 1$ for all $e \in E$, and for a pseudomarginal τ that belongs to the constraint set $\mathrm{LOCAL}(G)$ (so that, in particular, the marginalization condition $\sum_k \tau_{st;jk} = \tau_{s;j}$ holds at each edge). In this case, by expanding each mutual information term $I_{st}(\tau_{st})$ as the sum $H_s(\tau_s) + H_t(\tau_t) - H_{st}(\tau_{st})$, it can be seen that the function $-Q(\tau;\mathbf{1})$ is equivalent to the Bethe entropy approximation defined in equation (32).

It is important to note, however, the condition $\boldsymbol{\rho_e} = \mathbf{1}$ implies that each edge belongs to every spanning tree with probability one, a condition which holds if and only if the graph G is actually a tree. Therefore, for a graph with cycles, the vector $\boldsymbol{\rho_e} = \mathbf{1}$ is not a member of the spanning tree polytope $\mathbb{T}(G)$. The Bethe approximation for a graph with cycles entails making—at least from the perspective of Theorem 1—an *invalid* choice $\boldsymbol{\rho_e}$ of edge appearance probabilities. For any $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, Theorem 1 guarantees that the function $Q(\tau;\boldsymbol{\rho_e})$ is convex in terms of τ. Since $\mathbf{1} \notin \mathbb{T}(G)$ for a graph with cycles, this convexity result does not apply to the Bethe variational problem. Indeed, with the exception of certain special cases [see, e.g., 26], [29], the Bethe problem is typically not convex.

B. Optimization of the distribution $\vec{\boldsymbol{\rho}}$

Up to this point, the distribution $\vec{\boldsymbol{\rho}}$ over spanning trees has been held fixed; we now consider how to optimize this choice. Suppose that for a fixed distribution $\vec{\boldsymbol{\rho}}$, we have performed the optimization over $\tau \in \mathrm{LOCAL}(G)$ specified by equation (27) of Theorem 1. This optimization can be used to define the following function:

$$R(\boldsymbol{\rho_e};\bar{\theta}) \;\; := \;\; \max_{\tau\in\mathrm{LOCAL}(G)} \{\langle \tau,\, \bar{\theta}\rangle - Q(\tau;\boldsymbol{\rho_e})\} \qquad (33)$$

Note that R inherits a crucial property of Q: it depends on the spanning tree distribution $\vec{\boldsymbol{\rho}}$ only indirectly, via the vector of edge appearance probabilities $\boldsymbol{\rho_e}$. Therefore, in order to optimize the choice of spanning tree distribution $\vec{\boldsymbol{\rho}}$, it suffices to optimize R over $|E|$-dimensional vectors $\boldsymbol{\rho_e}$ in the spanning tree polytope $\mathbb{T}(G)$. Doing so leads to an upper bound that is jointly optimal over both τ and $\boldsymbol{\rho_e}$ (and hence $\vec{\boldsymbol{\rho}}$). The properties of this upper bound are summarized in the following result:

Theorem 2 (Jointly optimal bounds).
(a) The function R is convex in terms of $\boldsymbol{\rho_e}$. Moreover, minimizing it subject to $\boldsymbol{\rho_e} \in \mathbb{T}(G)$ yields an upper bound that is jointly optimal over both pseudomarginals τ and edge appearance probabilities $\boldsymbol{\rho_e}$:

$$\Phi(\bar{\theta}) \;\; \leq \;\; \min_{\boldsymbol{\rho_e}\in\mathbb{T}(G)} \; R(\boldsymbol{\rho_e};\bar{\theta}). \qquad (34)$$

(b) For any tree T in the support set $\mathrm{supp}(\vec{\boldsymbol{\rho}}^*)$ (i.e., for which $\rho(T) > 0$), the jointly optimal edge appearances $\boldsymbol{\rho_e}^*$ and pseudomarginals $\tau^* \equiv \tau^*(\boldsymbol{\rho_e}^*)$ satisfy the relation:

$$\sum_{(s,t)\in E(T)} I_{st}(\tau^*_{st}) = \sum_{(s,t)\in E} \rho^*_{st} I_{st}(\tau^*_{st}). \qquad (35)$$

(c) The jointly optimal value $\min_{\boldsymbol{\rho_e}\in\mathbb{T}(G)} \; R(\boldsymbol{\rho_e};\bar{\theta})$ of the upper bound is characterized by the following min-max relation:

$$\min_{\boldsymbol{\rho_e}\in\mathbb{T}(G)} \; \max_{\tau\in\mathrm{LOCAL}(G)} \big\{\langle \tau,\, \bar{\theta}\rangle - Q(\tau;\boldsymbol{\rho_e})\big\} \; = \; \max_{\tau\in\mathrm{LOCAL}(G)} \; \min_{\boldsymbol{\rho_e}\in\mathbb{T}(G)} \big\{\langle \tau,\, \bar{\theta}\rangle - Q(\tau;\boldsymbol{\rho_e})\big\}. \qquad (36)$$

Proof: See Appendix B.

Remarks: It is helpful, in order to gain further insight into parts (b) and (c) of Theorem 2, to adopt a game-theoretic perspective. We begin by developing an important consequence of equation (35). Given some spanning tree $T = (V, E(T))$, consider the negative entropy of the tree-structured distribution $p(\mathbf{x};\Pi^T(\tau^*))$:

$$\Phi^*(\Pi^T(\tau^*)) \;\; = \;\; -\sum_{s\in V} H_s(\tau^*_s) + \sum_{(s,t)\in E(T)} I_{st}(\tau^*_{st}) \qquad (37)$$

Note that the sum over single node entropy terms (i.e., $\sum_{s \in V} H_s$) in equation (37) is the same for any spanning tree. Moreover, equation (35) guarantees that the sum of mutual information terms in equation (37) is constant regardless of the choice of spanning tree $T \in \operatorname{supp}(\vec{\boldsymbol{\rho}})$. Consequently, the tree-structured negative entropy $\Phi^*(\Pi^T(\tau^*))$ is equal to the same constant for all spanning trees in $\operatorname{supp}(\vec{\boldsymbol{\rho}})$.

This entropy equalization can be understood by interpreting Theorem 2 from a game-theoretic perspective. In particular, let us consider a two-person zero-sum game, in which Player 1 chooses local pseudomarginals τ so as to maximize $\langle \tau, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})$, whereas Player 2 chooses a spanning tree so as to minimize the same quantity. The game is zero-sum, because any gain by either player translates into a loss for the other player. In this light, the minimax relation in equation (36) specifies the value of the game. With reference to the entropy equalization of Theorem 2(b), suppose that Player 1 chooses a pseudomarginal τ such that for two trees T_1, T_2, we have $\Phi^*(\Pi^{T_1}(\tau)) \;\; > \;\; \Phi^*(\Pi^{T_2}(\tau))$. From the game-theoretic perspective, any optimal strategy of Player 2 would exploit this discrepancy by reallocating more weight to tree T_1. Therefore, at an equilibrium point, the entropies of the tree-structured distributions should be equalized.

Example 3. To illustrate Theorem 2(a), we follow up on the single cycle graph of Example 2. For the choice of exponential parameter $\bar{\theta}$ given there, a symmetry argument establishes that the uniform edge appearance probabilities (i.e., $\rho_e = 3/4$ for all edges $e \in E$) will be optimal in the sense of Theorem 2(a), so that the upper bound computed in Example 2 is also *jointly* optimal.

Now suppose that we define a new exponential parameter $\widetilde{\theta}$ by setting $\widetilde{\theta}_\alpha = \bar{\theta}_\alpha$ for all indices $\alpha \in \mathcal{I}$, except for those indices corresponding to edge $(2,3)$ for which we set $\widetilde{\theta}_{23} = 3\,\bar{\theta}_{23}$. By solving the variational problem (27) with this exponential parameter and uniform ($\rho_e = 3/4$) edge weights, we obtain an upper bound 6.345, optimal in the sense of Theorem 1, on the exact value of the log partition function $\Phi(\widetilde{\theta}) \approx 6.333$. However, since the problem is no longer symmetric, the uniform choice of edge appearance will no longer be optimal. Therefore, it is appropriate to consider joint optimization over both τ and $\boldsymbol{\rho_e}$, as dictated by Theorem 2. Performing this optimization leads to the following optimal edge appearance probabilities:

$$\boldsymbol{\rho_e}^* \approx \begin{bmatrix} 0.92 & 0.54 & 0.54 & 1 \end{bmatrix}.$$

Note that the optimum assigns edge appearance probability of one to the edge $(2,3)$, which has the strongest interaction. As a result, this edge must appear in any spanning tree in the support of the optimizing distribution $\vec{\boldsymbol{\rho}}^*$. This set of edge appearance probabilities, combined with the associated $\tau^*(\boldsymbol{\rho_e}^*)$, yields the upper bound $R(\boldsymbol{\rho_e}^*; \widetilde{\theta}) \approx 6.339$ on the true log partition function $\Phi(\widetilde{\theta}) \approx 6.333$. Note that this upper bound is tighter than the previous bound (≈ 6.345) based on uniform edge appearance probabilities. □

V. Algorithms for optimization

In this section, we discuss algorithms for computing both the upper bound of Theorem 1, as well as the jointly optimal upper bound of Theorem 2. First of all, the optimization specified by Theorem 1 is a fairly standard convex program. Interestingly, one method for solving it iteratively is a tree-reweighted variant of the sum-product algorithm, as presented in Appendix C. Secondly, with reference to Theorem 2, the probability distribution $\vec{\boldsymbol{\rho}}$ itself consists of a number for each of the $|\mathfrak{T}(G)|$ spanning trees of G, and so will be too large to optimize directly. As noted earlier, the key property is that the function R of Theorem 2 depends on $\vec{\boldsymbol{\rho}}$ only indirectly, via the lower dimensional vector $\boldsymbol{\rho_e}$ of edge appearance probabilities. It turns that out the optimal choice of edge appearance probabilities can be found by an iterative algorithm, each step of which entails solving a maximum-weight spanning tree problem.

A. Optimizing the pseudomarginals

We first consider how to compute the upper bound of Theorem 1. More specifically, for a fixed $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, we want to solve the problem:

$$\max_{\tau \in \text{LOCAL}(G)} \big\{ \langle \tau, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e}) \big\}. \tag{38}$$

Note that the objective function of this maximization problem is concave, and the constraints are linear. Consequently, a variety of standard methods from nonlinear programming can be used [6]. As noted earlier in Section IV-A.3, this optimization problem is closely related to the Bethe variational problem [46], and hence to the sum-product algorithm. This link suggests that it should be possible to solve the optimization problem (38) by a type of distributed "message-passing" algorithm. Indeed, we now describe a form of *tree-reweighted sum-product* algorithm, analogous to but distinct from the usual sum-product updates, that can be be used to solve the optimization problem (38).

As with the sum-product updates, our algorithm also involves passing messages from node to node. We let $M_{st}(x_t)$ denote the message passed from node s to node t; it is a vector of length m_t, with one component for each value of $x_t \in \mathcal{X}_t$. For compactness in notation, we

Algorithm 1 (Tree-reweighted sum-product).

1) *Initialize the messages* $\mathbf{M}^0 = \{M^0_{st}\}$ *with arbitrary positive real numbers.*
2) *For iterations* $n = 0, 1, 2, \ldots$, *update the messages as follows:*

$$M^{n+1}_{ts}(x_s) \quad = \quad \alpha \sum_{x'_t \in \mathcal{X}_t} \left\{ \exp\left(\frac{\bar{\theta}_{st}(x_s, x'_t)}{\rho_{st}} + \bar{\theta}_t(x'_t) \right) \frac{\prod_{v \in \Gamma(t) \backslash s} \left[M^n_{vt}(x_t) \right]^{\rho_{vt}}}{\left[M^n_{st}(x_t) \right]^{(1-\rho_{ts})}} \right\}. \qquad (39)$$

Fig. 5. Parallel edge-based form of the tree-reweighted sum-product algorithm. The quantity $\alpha > 0$ denotes a normalization constant (typically chosen such that $\sum_{x_s} M^{n+1}_{ts}(x_s) = 1$).

let $\bar{\theta}_t(x_t)$ be the function that takes the value $\bar{\theta}_{t;k}$ when $x_t = k$; explicitly, this function is defined as follows:

$$\bar{\theta}_t(x_t) \quad = \quad \sum_{k \in \mathcal{X}_t} \bar{\theta}_{t;k} \delta_{t;k}(x_t).$$

The function $\bar{\theta}_{st}(x_s, x_t)$ is defined similarly.

With this notation, the algorithm takes the form shown in Figure 5. Observe that the updates (39), although quite similar to the standard sum-product updates [46], differs in some key ways. First of all, the weights $\theta_{st;jk}$ corresponding to edge (s,t) are all rescaled by $1/\rho_{st}$. Secondly, all the messages M_{us} for nodes $u \in \Gamma(s)\backslash t$ are exponentiated by the corresponding edge appearance ρ_{us}. Lastly, the message M_{st} running in the *reverse direction* on edge (s,t) is involved in updating M_{ts}. Despite these differences, it is still possible to perform the message updates in a parallel fashion, as in the parallel form of sum-product. It is also possible to perform reparameterization updates over spanning trees [see 35], [37].

Overall, the complexity of performing the tree-reweighted updates (39) is identical to the standard sum-product updates. In Appendix C, we prove that the tree-reweighted updates always have a unique fixed point that specifies the global optimum of problem (38). Although we have not performed a thorough convergence analysis at this point, we have observed empirically that the updates converge if they are suitably damped,

B. Optimizing the spanning tree distribution

We now consider the problem of solving the problem $\min_{\boldsymbol{\rho_e} \in \mathbb{T}(G)} R(\boldsymbol{\rho_e}; \bar{\theta})$, as required to compute the upper bound of Theorem 2. A challenge to be overcome here lies in the nature of the spanning tree polytope $\mathbb{T}(G)$. For a general graph, the number of linear constraints required to specify this polytope grows exponentially ($\mathcal{O}(2^N)$) with the graph size [14], [10]. This exponential growth precludes the use of any method that deals directly with the constraints themselves. Interestingly, it turns out that despite the exponential number of constraints characterizing $\mathbb{T}(G)$, maximizing a linear function over this polytope is feasible. Indeed, this task is equivalent to solving a *maximum weight spanning tree* problem, which can be solved by well-known greedy methods (e.g., Kruskal's algorithm [20]).

The feasibility of solving a linear program over the spanning tree polytope suggests the use of the *conditional gradient method* [6]. In application to the problem $\min_{\boldsymbol{\rho_e} \in \mathbb{T}(G)} R(\boldsymbol{\rho_e}; \bar{\theta})$, each step of the conditional gradient method entails finding the descent direction by solving the optimization problem:

$$\widetilde{\boldsymbol{\rho_e}}^{n+1} \quad = \quad \arg \min_{\boldsymbol{\rho_e} \in \mathbb{T}(G)} \left[\nabla R(\boldsymbol{\rho_e}^n; \bar{\theta}) \right]^T \left[\boldsymbol{\rho_e} - \boldsymbol{\rho_e}^n \right] \qquad (40)$$

Since the set $\mathbb{T}(G)$ is a polytope and the cost function $\left[\nabla R(\boldsymbol{\rho_e}^n; \bar{\theta}) \right]^T \left[\boldsymbol{\rho_e} - \boldsymbol{\rho_e}^n \right]$ is linear in $\boldsymbol{\rho_e}$, finding this descent direction corresponds to solving a linear program over the spanning tree polytope. The optimum of a linear program is always attained at (at least one of) the vertices of the constraint polytope. In the case of the spanning tree polytope, each of these vertices corresponds to the indicator vector of a particular spanning tree. (See the definition of $\mathbb{T}(G)$ in equation (13).) Moreover, note that this linear program can be interpreted as a minimum weight spanning tree problem, in which the element of the gradient $\frac{\partial R}{\partial \rho_{st}}(\boldsymbol{\rho_e}^n; \bar{\theta})$ serves as the weight on edge (s,t). As a consequence, the descent direction can be computed very efficiently via Kruskal's algorithm for finding optimal weighted spanning trees [20]. With the use of suitable data structures, the computational complexity of Kruskal's algorithm is $\mathcal{O}(|E| \log N)$, where $|E|$ is the number of edges and $N = |V|$ is the number of vertices in the graph.

To completely specify the algorithm, it remains to compute the gradient vector $\nabla R(\boldsymbol{\rho_e}^n; \bar{\theta})$. For a given $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, let $\tau^*(\boldsymbol{\rho_e})$ denote the optimum of the variational problem (38). It can be shown (see Appendix B) that elements of the gradient $\frac{\partial R}{\partial \rho_{st}}$ are given by negative mutual information terms $-I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*))$. We let $I(\tau^*(\boldsymbol{\rho_e}^n))$ denote a vector formed of these mutual information terms, one for each edge. So as to facilitate subsequent interpretation, instead of solving the *minimum* spanning tree spanning problem with

Algorithm 2 (Conditional gradient). 1) *Initialize at a valid* $\boldsymbol{\rho_e}^0 \in \mathbb{T}(G)$.

2) *For iterations* $n = 0, 1, 2, \ldots$, *solve a maximum weight spanning tree problem to find the descent direction:*

$$\widetilde{\boldsymbol{\rho_e}}^{n+1} = \arg \max_{\boldsymbol{\rho_e} \in \mathbb{T}(G)} \langle I(\tau^*(\boldsymbol{\rho_e}^n)),\, \boldsymbol{\rho_e} - \boldsymbol{\rho_e}^n \rangle\}.$$

3) *Form* $\boldsymbol{\rho_e}^{n+1} = (1-\alpha^n)\boldsymbol{\rho_e}^n + \alpha^n \widetilde{\boldsymbol{\rho_e}}^{n+1}$ *where* $\alpha^n \in (0,1)$ *is a step size parameter.*

Fig. 6: Conditional gradient method for optimizing the choice of edge appearance probabilities.

$-I(\tau^*(\boldsymbol{\rho_e}^n))$, we solve the *maximum* weight spanning tree problem with the non-negative mutual information vector $I(\tau^*(\boldsymbol{\rho_e}^n))$. The algorithm then takes the form shown in Figure 6.

With an appropriate choice of step size α^n at each iteration (chosen, for instance, by Armijo's rule), the conditional gradient updates of Algorithm 2 are guaranteed to converge to the global minimum [6].

Remark: The second step of Algorithm 2 has an interesting interpretation in terms of fitting a tree-distribution to a collection of data. In particular, for a given set of data $\{\mathbf{y}^1, \ldots, \mathbf{y}^n\}$, let $\widehat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \delta(\mathbf{x} = \mathbf{y}^i)$ denote the associated empirical distribution. Now consider the problem of finding the tree-structured distribution p_{tree} that best fits the data in the sense of minimizing the Kullback-Leibler divergence $D(\widehat{p} \parallel p_{tree})$. Chow and Liu [11] showed that an optimal tree-structured distribution (which need not be unique) can be obtained by solving a maximum weight spanning tree problem. In particular, the edge weights are specified by mutual information terms $I_{st}(\widehat{\mu}_{st})$, where $\widehat{\mu}_{st}$ is the empirical marginal distribution on (x_s, x_t) defined by $\widehat{p}$. If we view the pseudomarginals $\tau^*(\boldsymbol{\rho_e}^n)$ at iteration n as such a set of empirical marginals, then each iteration of Algorithm 2 entails pushing the edge appearance vector $\boldsymbol{\rho_e}$ in the direction of the spanning tree that best fits the current data in the Kullback-Leibler sense.

C. Experiments

In this section, we present the results of applying the previously described algorithms to compute the upper bounds specified in Theorems 1 and 2, as applied to different graphs and varied types of interactions. In preliminary versions of this work [35], [36], we presented experimental results for relatively weak couplings. Here we show the results of simulations for a binary-valued random vector on both lattices and fully-connected graphs over a much wider range of couplings, from relatively weak to very strongly coupled. Over this range, we compare the upper bounds given by our techniques to lower bounds on the log partition function obtained from the naive mean field method [19]. For the sake of comparison, we also plot the value of the optimized Bethe free energy [46]; however, it should be kept in mind that in general, the Bethe free energy *does not provide a bound*—either lower or upper—on the log partition function. Moreover, we compare the accuracy in the pseudomarginals obtained from the convex upper bounds to those obtained from the Bethe approximation and the ordinary sum-product algorithm.

The connectivity of the 2D grid is sufficiently sparse that the use of tree approximations is reasonable. For the fully connected graph, in contrast, approximations based on trees are less likely to perform well. Our purpose in showing results for the fully connected case, then, is to demonstrate how approximations based on sparse graphs, such as those of this paper, may behave poorly for very densely connected problems.

1) Procedure: For each trial, we defined a distribution $p(\mathbf{x}; \bar{\theta})$ by a random choice of exponential parameter vector $\bar{\theta}$. We used the standard Ising representation given in equation (2), except that each random variable x_s took values in $\{-1, +1\}$. This so-called "spin" representation turns out to be more convenient than a $\{0,1\}$-representation for defining attractive and mixed interactions. In all cases, we choose $\bar{\theta}_s \sim \mathcal{U}[-0.05, 0.05]$ independently for each $s \in V$. For a given edge strength $\omega > 0$, we set the pairwise potentials in one of two ways:

(a) for *attractive* interactions, we set $\bar{\theta}_{st} \sim \mathcal{U}[0, \omega]$ independently for each edge $(s,t) \in E$

(b) for *mixed* interactions, we set $\bar{\theta}_{st} \sim \mathcal{U}[-\omega, \omega]$ independently for each edge $(s,t) \in E$

For each of the graphs and each of the two edge coupling conditions (attractive or mixed), we ran simulations with edge strengths ω ranging from 0 to 2. The inner optimization (i.e., solving the problem $\max_{\tau \in \mathrm{LOCAL}(G)} \{\langle \bar{\theta},\, \tau \rangle - Q(\tau; \boldsymbol{\rho_e})\}$) was performed using the tree-weighted sum-product algorithm, as described in Appendix C, with damping factor $\gamma = 0.4$. The optimization of the edge appearance probabilities $\boldsymbol{\rho_e}$ was performed with the conditional gradient method (Algorithm 2), where the step size choice was made by the Armijo rule [6]. We computed the value of the actual partition function $\Phi(\bar{\theta})$ either by brute force enumeration (fully-connected graphs), or by forming a junction tree and then performing exact computations (lattices). The mean field lower bounds were computed using the standard mean field co-ordinate ascent algorithm [39], taking the best optimum from a randomly chosen starting point, and the uniform starting point. Finally, we computed the Bethe free energy by first applying the sum-product algorithm with damping factor $\gamma = 0.4$, and then switching to a convergent double-loop alternative [15] if ordinary sum-product failed to converge.

In terms of computational complexity, our tree-reweighted updates involve a computational cost per update that is equivalent to the ordinary sum-product algorithm (i.e., $\mathcal{O}(|E|)$ per update). The tree-reweighted algorithm has a lower complexity than the double-loop algorithm that needs to be used if the ordinary sum-product updates do not converge. It should be noted that the complexity of the mean field algorithm is $\mathcal{O}(N)$, so that it is cheaper than either the ordinary or reweighted sum-product updates.

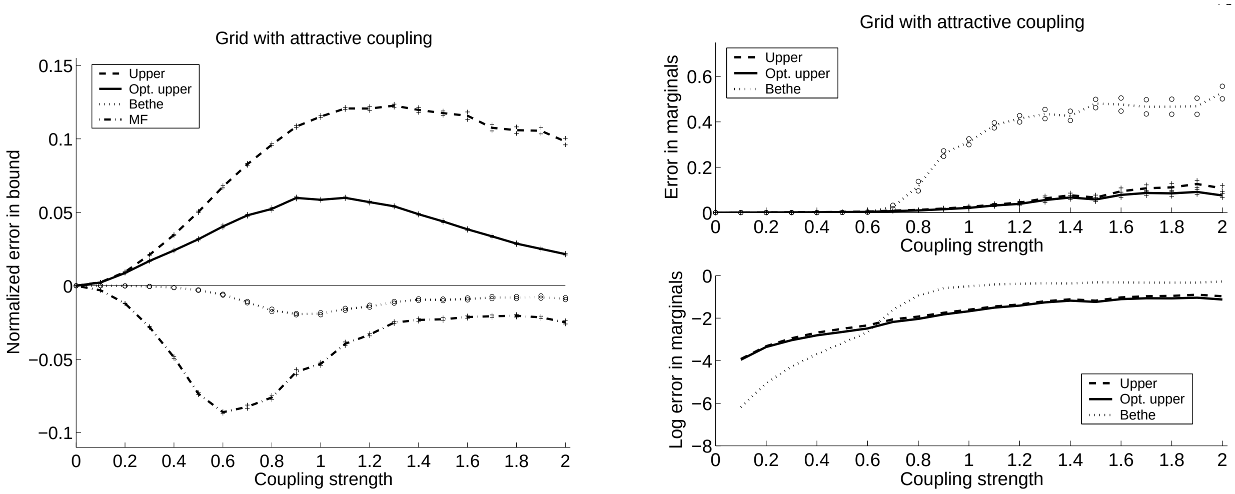
2) Square lattices: We first show the results of simulations for square lattices in two dimensions with four nearest-neighbor connectivity. Although our simulations were necessarily limited to grids with $N = 100$ nodes (due to the computational complexity of performing exact calculations), it should be noted that as with the ordinary sum-product algorithm, the algorithms for solving the optimization problems given in Theorems 1 and 2 can be applied to much larger problems. We performed 20 trials for each of the two conditions (attractive or mixed) and each setting of the edge strength in increments of 0.1 in the interval $[0, 2]$.

Shown in Figure 7(a) and Figure 8(a) are plots of the average *normalized error* $[\text{Bound} - \Phi(\bar{\theta})]/N$ versus the edge strength ω over the interval $[0, 2]$ in the cases of attractive and mixed couplings respectively. Here the terminology "Bound" denotes either an upper bound (based on the convex approximations), or a lower bound (mean field). For the Bethe free energy, we plot the average of the negative absolute value—namely, the quantity $-|\,\text{Bethe} - \Phi(\bar{\theta})|/N$. For this reason, it appears as a lower bound (but it is neither a lower nor upper bound in general). Each panel (a) displays the relative error in two types of upper bounds. The "unoptimized" curve shows the bound of Theorem 1 with the fixed choice of uniform edge appearance probabilities $\rho_e = (N-1)/|E|$, whereas the "optimized" curve corresponds to the jointly optimal (over both τ and $\vec{\boldsymbol{\rho}}$) upper bounds of Theorem 2. On the other hand, Figure 7(b) and Figure 8(b) show the average ℓ_1-error $\frac{1}{N}\sum_{i=1}^{N} |\tau_i - \mu_i|$ between the pseudomarginal τ_i and the true marginal probability at node i, averaged over all nodes. Here we compared the pseudomarginals obtained either from the "unoptimized" upper bound, the "optimized" upper bound, and the ordinary sum-product pseudomarginals.

Consider first the case of attractive couplings, as illustrated in Figure 7. With reference to the bounds plotted in panel (a), we see that the upper bounds are relatively tight for low edge strengths, and their tightness decreases as the edge strength is increased over a certain intermediate range of coupling strengths. Ultimately, for large couplings, the bounds again become tighter as the edge strength is increased. Indeed, for this case of pure attractive couplings, it can be shown that the bounds become tight as the coupling strength ω tends to infinity [39]. In comparison to mean field, the unoptimized upper bound is slightly worse, whereas the optimized upper bound is slightly better. Optimizing the edge appearance probabilities can lead to significantly better upper bounds. This effect is especially pronounced as the edge strength is increased, in which case the distribution of edge weights $\bar{\theta}_{st}$ becomes more inhomogeneous. As might be expected, all of the bounds (both lower and upper) are, at least in general, worse than the Bethe approximation, which does not provide a bound. Shown in Figure 7(b) is the comparison of the average ℓ_1 error in the pseudomarginal approximations of the convex methods versus the ordinary sum-product algorithm. The upper plot shows the error on a standard scale, and the bottom on a logarithmic scale. Here we see that the ordinary sum-product is more accurate for edge couplings below $\omega \approx 0.6$. Beyond this point, however, the accuracy of the sum-product algorithm degrades quite rapidly, whereas the pseudomarginals from the convex approximation remain quite accurate.

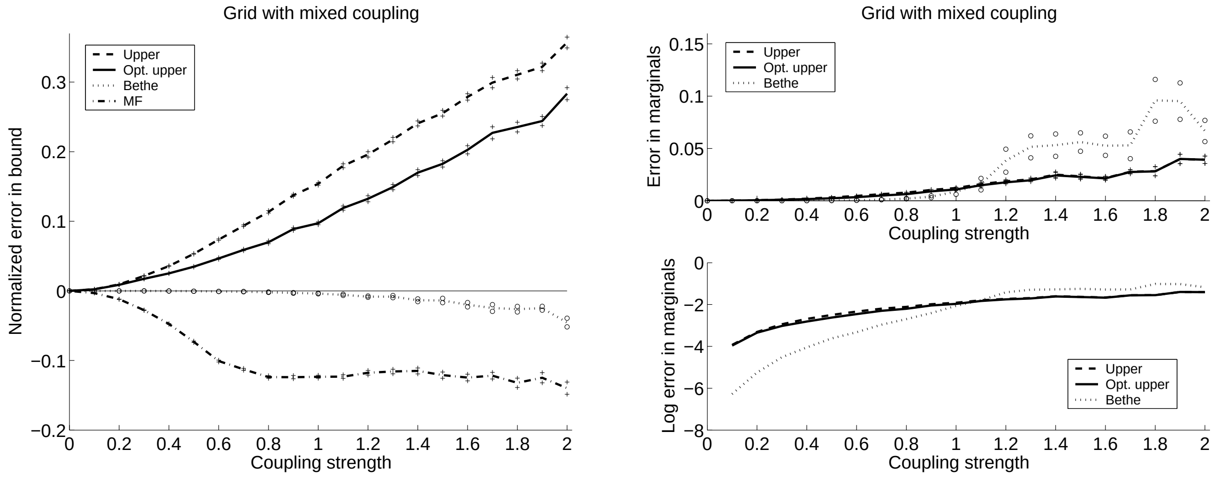
Now let us turn to the case of mixed couplings, as illustrated in Figure 8. Looking at the comparison among bounds in panel (a), we again see that the bounds are tightest for weak couplings, and their accuracy degrades as the coupling strength is increased. In the regime of relatively weak couplings, the upper bounds are superior to mean field, but this advantage is lost as the coupling strength is increased. Note that neither the upper nor the lower bounds are as accurate as the Bethe approximation. A notable difference between mixed couplings versus attractive couplings is that the accuracy of the upper bounds does not eventually improve as the coupling strength is increased. Turning to the comparison of pseudomarginals in panel (b), we see a similar pattern to the attractive case. In the regime of weak couplings, the ordinary sum-product algorithm is superior, whereas for sufficiently strong couplings, the pseudomarginals from the convex upper bounds appear to be more accurate.

It is worthwhile emphasizing the importance of the dual formulation of our bounds. Indeed, the naive approach of attempting to optimize the primal formulation of the bounds (i.e., as in equation (17)) would require dealing with the astronomical number $\approx 5.69 \times 10^{42}$, corresponding to the number of spanning trees on the



(a) (b)

Fig. 7. Comparisons of convex upper bounds, mean field (MF), and the Bethe approximation for attractive couplings on a grid-structured model. (a) Comparison of the normalized error in the bounds (and Bethe approximation) to the log partition function. (b) Comparison of the error in the approximate marginals.

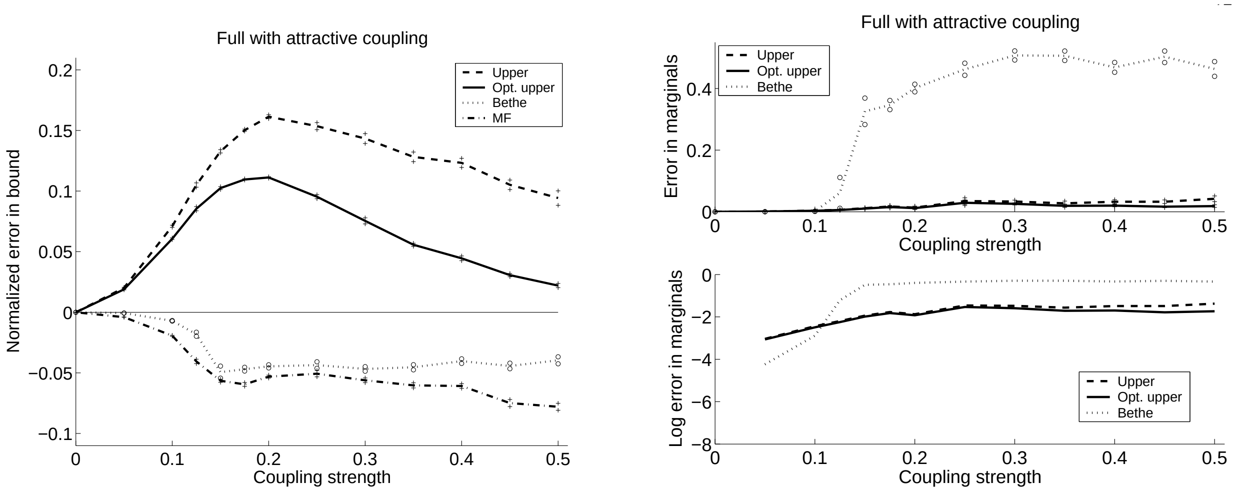


(a) (b)

Fig. 8. Comparisons of convex upper bounds, mean field (MF), and the Bethe approximation for mixed couplings on a grid-structured model. (a) Comparison of the normalized error in the bounds (and Bethe approximation) to the log partition function. (b) Comparison of the error in the approximate marginals.

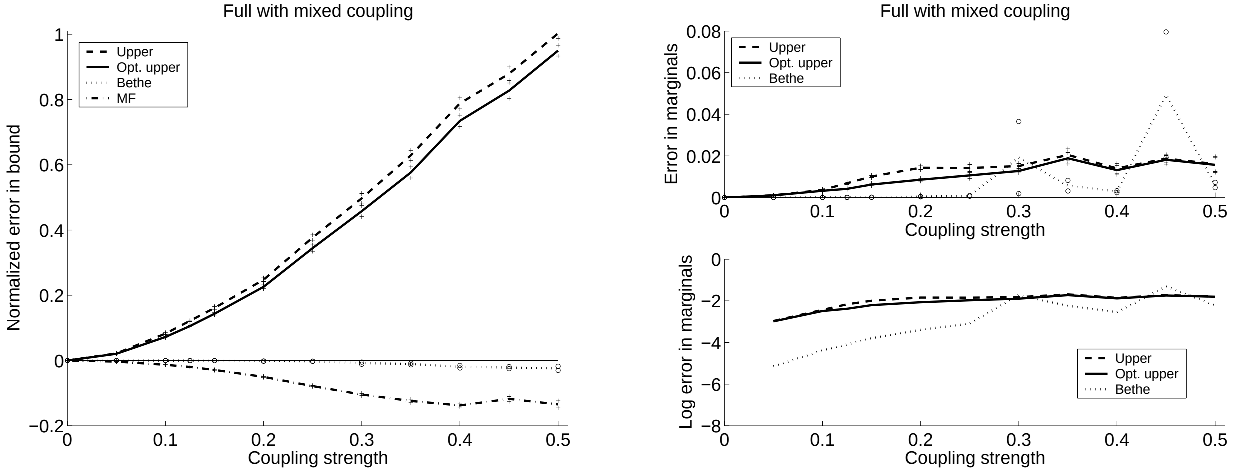
grid with $N = 100$ nodes. (This number can be calculated by applying the Matrix-Tree theorem [7].)

3) Fully connected graphs: To provide contrast with the relatively sparse case of a 2D grid, we also performed simulations on a very dense graph — in particular, the fully connected graph K_{16} on $N = 16$ nodes, with edge strengths ω ranging from 0 to 0.5. The results, with the same legend and lay-out as the lattice simulations, are shown in Figures 9 and 10 for the cases of attractive and mixed couplings respectively. With references to panels (a) in Figures 9 and 10, overall the upper bounds are less accurate than their counterparts on the lattices; nonetheless, the basic qualitative pattern of results is preserved. In the case of attractive couplings in Figure 9(a), we see that the tightness of the bounds degenerates up until a certain point, and then starts to improve again. For the case of mixed couplings in Figure 10(a), the accuracy decreases steadily (in a roughly linear fashion) as the coupling strength is increased. For this case of mixed couplings, the upper bounds are significantly weaker than either the mean field lower bound or the Bethe approximation. Turning now to panel (b) in each of the two figures, the plots of the errors in the pseudomarginals show a similar pattern. Looking first at the attractive cou-



(a) (b)

Fig. 9. Comparisons of convex upper bounds, mean field (MF), and the Bethe approximation for attractive couplings on a fully connected graph. (a) Comparison of the normalized error in the bounds (and Bethe approximation) to the log partition function. (b) Comparison of the error in the approximate marginals.



(a) (b)

Fig. 10. Comparisons of convex upper bounds, mean field (MF), and the Bethe approximation for mixed couplings on a fully connected graph. (a) Comparison of the normalized error in the bounds (and Bethe approximation) to the log partition function. (b) Comparison of the error in the approximate marginals.

pling case in Figure 9(b), we see that the ordinary sum-product approximation is more accurate in the regime of weak couplings, but rapidly deteriorates beyond a certain coupling strength. Over this same range of coupling strengths, the accuracy of the pseudomarginals from the convex approximations remains surprisingly robust. The accuracy of the different types of pseudomarginals in the mixed case (Figure 10(b)) is qualitatively similar, though not as clear cut.

D. Scaling with problem size

Finally, we investigate how our bounds scale with the number of nodes $N = |V|$. In most cases, the parameters θ are scaled such that the log partition function scales linearly with N; thus, we also expect that our bounds should scale linearly in N. The tree-reweighted message-passing updates scale well to larger problem sizes; however, so as to enable comparison to the exact answer, we focus on an exactly-solved model: namely, the Ising model on the toric grid with homogeneous positive interactions (i.e, $\theta_{st} = J > 0$ for all edges $(s,t) \in E$), and no observations (i.e., $\theta_s = 0$ for all

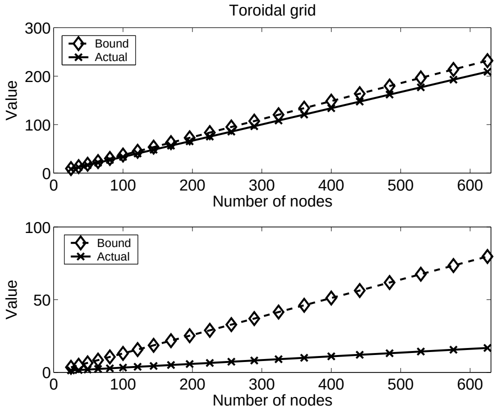


Fig. 11. Scaling of the bounds as a function of the number of nodes N in a 4-nearest-neighbor grid with toroidal boundary conditions and homogeneous interactions between neighbors. (a) Case of relatively weak coupling $J = 0.2$ in the spin representation of the Ising model. (b) Case of stronger coupling ($J = 0.5$).

$s \in V$). The high degree of node and edge symmetry simplifies its analysis. Each of the N nodes has exactly 4 neighbors, so that there are $|E| = 2N$ edges in total. In ground-breaking work, Onsager [28] showed how to reduce the computation of partition function in this model to a series of eigenvalue computations, so that it can be computed exactly for large models.

With reference to our upper bounds, it also follows as a consequence of the symmetry that the optimal choice (in the sense of Theorem 2) of edge appearances is the uniform one $\bar{\rho} = \frac{|V|-1}{|E|} = \frac{N-1}{2N}$. Figure 11 compares the exact results obtained from Onsager's method to the tree-reweighted upper bound for a range of graph sizes (N), and two different coupling strengths. The top panel (a) shows the case of relatively weak coupling, for which the bound remains quite accurate even for relatively large problems. For the stronger couplings illustrated in (b), in contrast, the accuracy of the bound is seen to degrade more swiftly. In both cases, the upper bound scales linearly with N, which is consistent with the behavior of the exact log partition function.

E. *Related experimental results*

We conclude our experimental section by describing some related experimental results. Wiegerinck and Heskes [44] have shown that optimizing reweighted Bethe free energies, though not necessarily convex, can lead to better results in approximate inference. In their work, they proposed a heuristic procedure for optimizing weights on each edge. These weights are analogous to our edge appearance probabilities, but are not required to belong to the spanning tree polytope (and hence the guarantee of convexity is lost). Interestingly, they showed that their method of adjusting the weights can in many cases lead to better results than belief propagation. In later work, Wiegerinck [43] performed experimental comparisons of tree-reweighted belief propagation and standard belief propagation, with results analogous to those shown here, as well as in-depth comparisons to reweighted variants of generalized belief propagation (as discussed in the following section).

VI. Generalization to higher order MRFs

Our development in the previous sections focused on the case of a pairwise Markov random field (MRF), for which the collection ϕ of potentials involves only singleton and edge functions. For many problems of interest, the associated MRF includes potential functions over cliques of size three or larger. In this section, we briefly outline how the analysis described in the preceding sections can be generalized to this case. In order to discuss MRFs with higher order cliques, it is convenient to introduce the formalism of hypergraphs [e.g., 5]. Having set up this machinery, we demonstrate how *hypertrees*, which represent a natural generalization of trees, can be used to derive upper bounds on the log partition function. Finally, we demonstrate using a simple example the link between convex combinations of hypertrees and "convexified" forms of Kikuchi free energies [46], [47].

A. *Hypergraphs*

Hypergraphs are a natural generalization of graphs. In particular, a hypergraph $G_{\text{HYP}} = (V, E)$ consists of a vertex set $V = \{1, \ldots, N\}$, and a set of hyperedges E, where each *hyperedge* h constitutes a particular subset of V (i.e., an element of the power set of V). The set of hyperedges is a particular case of a partially-ordered set [33], for which the partial order is specified by the inclusion relation. Given two hyperedges $g, h \in E$, one of three possibilities can hold: (a) either g is contained within h, in which case we write $g < h$, or alternatively, (b) g contains h, which we denote by $g > h$, or (c) neither containment relation holds, in which case we say that g and h are incomparable. A hyperedge is *maximal* if it is not contained within any other hyperedge. We assume that the intersection between every pair of maximal hyperedges is contained within the hypergraph.

Given any hyperedge h, we define the sets of its *descendants* and *ancestors* in the following way:

$$\mathcal{D}(h) \quad = \quad \{g \in E \mid g < h\}, \qquad (41a)$$

$$\mathcal{A}(h) \quad = \quad \{g \in E \mid g > h\}. \qquad (41b)$$

With these definitions, an ordinary graph is a special case of a hypergraph, in which each maximal hyperedge

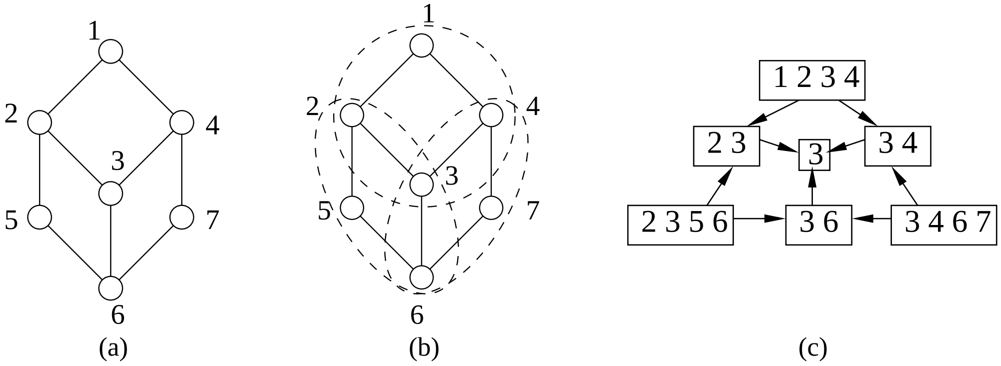


Fig. 13. Hypergraphs arise naturally by clustering operations. (a) Original pairwise Markov random field. (b) Particular clustering of the nodes. (c) Hypergraph defined by the clusters, as well as intersections between clusters, and intersections of intersections (node 3).

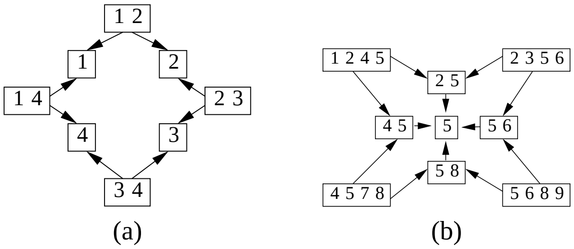


Fig. 12. Graphical representations of hypergraphs. Subsets of nodes corresponding to hyperedges are shown in rectangles, whereas the arrows represent inclusion relations among hyperedges. (a) An ordinary single cycle graph represented as a hypergraph. (b) A more complex hypergraph that does not correspond to an ordinary graph.

consists of a pair of vertices (i.e., an ordinary edge of the graph).[7]

A convenient graphical representation of a hypergraph is in terms of a diagram of its hyperedges, with (directed) edges representing the inclusion relations, as illustrated in Figure 12. As a special case, any ordinary graph can be drawn as a hypergraph; in particular, panel (a) shows the hypergraph representation of a single cycle on four nodes. Shown in panel (b) is a more complex hypergraph that does not correspond to an ordinary graph. To illustrate the notions of descendants and ancestors, the descendant set $\mathcal{D}(1245)$ is given by $\{(25), (45), (5)\}$, whereas $\mathcal{A}(5) = E \backslash (5)$ since every hyperedge is an ancestor of (5).

Even when the original problem of interest corresponds to a pairwise MRF, hypergraphs can arise naturally by clustering together nodes from the original graph. As an illustration, consider the pairwise Markov random field in Figure 13(a). One clustering of the nodes, obtained by grouping together nodes within 4-cycles, is shown in panel (b). We can use this clustering to define a set of hyperedges, formed by the clusters themselves (e.g., (1234)), intersections between the clusters (e.g., (23)), and the intersections between intersections (e.g., (3)). The resulting hypergraph is illustrated in panel (c). This particular procedure for constructing a hypergraph from a given graph is referred to as the Kikuchi method by Yedidia et al. [46], [47].

B. Hypertrees

Of particular importance are acyclic hypergraphs, which are also known as hypertrees. In order to define these objects, we require the notions of tree decomposition and running intersection, which are well-known in the context of junction trees [see 24], [13]. Given a hypergraph G_{HYP}, a *tree decomposition* is an acyclic graph in which the nodes are formed by the maximal hyperedges of G_{HYP}. Any intersection $g \cap h$ of two maximal hyperedges that are adjacent in the tree is known as a *separator set*. The tree decomposition has the *running intersection property* if for any two nodes g and h in the tree, all nodes on the unique path joining them contain the intersection $g \cap h$; such a tree decomposition is known as a *junction tree*.

A hypergraph is *acyclic* if it possesses a tree decomposition with the running intersection property. (Recall that we assume that any intersection between maximal hyperedges belongs to the hypergraph). The *width* of an acyclic hypergraph is the size of the largest hyperedge minus one; we use the term *k-hypertree* to mean a singly-connected acyclic hypergraph of width k. A *hyperforest* is a disjoint union of hypertrees. A hypertree is spanning if each vertex is contained within at least one hyperedge.

A simple illustration is provided by any tree of an ordinary graph: it is a 1-hypertree, because its maximal

[7] It should be noted there is a minor inconsistency between our definition of a hypergraph edge set, and a graph edge set; for hypergraphs, the set of hyperedges can include the individual vertices (unlike the corresponding edge set for an ordinary graph).

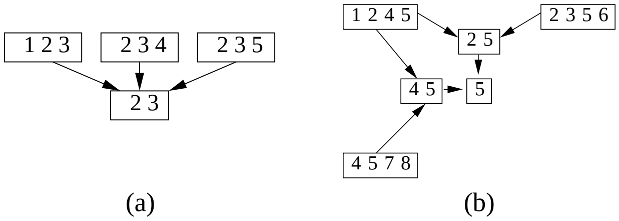


Fig. 14. Two examples of acyclic hypergraphs or hypertrees. (a) A hypertree of width two. The hyperedge (23) will appear twice as a separator set in any tree decomposition. (b) A hypertree of width 3. Hyperedges (25) and (45) are separator sets, and node 5 plays no role in the tree decomposition.

hyperedges (i.e., ordinary edges) all have size two. As a second example, the hypergraph of Figure 14(a) has maximal hyperedges of size three. This hypergraph is acyclic with width two, since it is in direct correspondence with the junction tree formed by the three maximal hyperedges, where (23) appears twice as a separator set. Figure 14(b) shows another hypertree; it is of width three, since its maximal hyperedges consist of four nodes. The junction tree in this case is formed of the three maximal hyperedges, using the two hyperedges of size two (i.e., (25) and (45)) as separator sets. In this case, the hyperedge (5) in the hypergraph diagram plays no role, since it is neither a maximal clique nor a separator set in the junction tree representation.

C. Hypertree factorization and entropy decomposition

Fundamental to the Lagrangian reformulation in our earlier work was the factorization of a tree-structured distribution in terms of local marginal distributions, as in equation (21). We now consider the hypertree analog of this decomposition. Given any hyperedge E in the hyperedge set $E(T)$ of a hypertree T, let $\tau_h(\mathbf{x}_h)$ denote the associated marginal distribution over the subset of variables $\mathbf{x}_h = \{x_s \mid s \in h\}$. Let $E_{\max}$ and E_{sep} denote the set of maximal hyperedges and separator sets in a junction tree associated with T. With this notation, we have the following well-known junction tree factorization [13], [24]:

$$p(\mathbf{x}) \quad = \quad \frac{1}{Z} \, \frac{\prod_{h \in E_{\max}} \tau_h(\mathbf{x}_h)}{\prod_{g \in E_{\text{sep}}} \left[\tau_g(\mathbf{x}_g)\right]^{d(g)-1}}. \qquad (42)$$

Here $d(g)$ denotes the number of maximal hyperedges adjacent to the separator set g in the hypertree.

It turns out to be more convenient for our purposes to make use of an alternative form of this factorization, which we describe here. Using the local marginals $\{\tau_h\}$, we define a function φ_h for *every* hyperedge $h \in E$ as follows:

$$\varphi_h(\mathbf{x}) \quad := \quad \frac{\tau_h(\mathbf{x}_h)}{\prod_{g \in \mathcal{D}(h)} \varphi_g(\mathbf{x}_g)}, \qquad (43)$$

where the set of descendants $\mathcal{D}(h)$ is defined in equation (41a). This definition is closely tied to the Möbius function associated with the poset of hyperedges [33]. With this definition, the hypertree factorization of $p(\mathbf{x})$ is very simple:

$$p(\mathbf{x}) \quad = \quad \prod_{h \in E(T)} \varphi_h(\mathbf{x}_h). \qquad (44)$$

We illustrate the hypertree factorization with a few examples:

Example 4 (Hypertree factorization).
(a) First suppose that the hypertree is an ordinary tree, in which case the hyperedge set consists of the union of the vertex set with the (ordinary) edge set. For any vertex s, we have $\varphi_s(x_s) = \tau_s(x_s)$, whereas for any edge (s,t) we have $\varphi_{st}(x_s, x_t) = \tau_{st}(x_s, x_t)/[\tau_s(x_s) \; \tau_t(x_t)]$. Therefore, in this special case, equation (44) reduces to the ordinary tree factorization of equation (21).

(b) Now consider the hypertree in Figure 14(a). First of all, we have $\varphi_{23} = \tau_{23}$, where we have omitted the explicit dependence on $\mathbf{x}$ for notational simplicity. Secondly, we compute $\varphi_{123} = \tau_{123}/\tau_{23}$, with similar expressions for φ_{234} and φ_{235}. Forming the product $\prod_{h \in E} \varphi_h$ yields the factorization that would be obtained from the junction tree representation in equation (42) — namely, $\tau_{123}\tau_{234}\tau_{235}/\tau_{23}^2$.

(c) As a third example, consider the hypertree of Figure 14(b). It is straightforward to calculate $\varphi_5 = \tau_5$, and $\varphi_{25} = \tau_{25}/\tau_5$ with a similar expression for φ_{45}. Next we calculate $\varphi_{2356} = \tau_{2356}/\varphi_{25}\varphi_5 = \tau_{2356}/\tau_{25}$, with an analogous expression for φ_{4578}. Finally, we calculate $\varphi_{1245} = \tau_{1245}/\varphi_{25}\varphi_{45}\varphi_5 = \tau_{1245}\tau_5/[\tau_{25}\tau_{45}]$. In this case, taking the product $\prod_{h \in E} \varphi_h$ over hyperedges leads to the expression

$$\frac{\tau_{1245}\tau_{4578}\tau_{2356}}{\tau_{25}\tau_{45}},$$

which (once again) agrees with the factorization that would be obtained by the junction tree representation (42). Note how the marginal τ_5 plays no role in the factorization, since it is neither a maximal hyperedge nor a separator set in the tree decomposition. □

D. Upper bounds based on hypertrees

We now describe the analogs for hypertrees of the basic upper bound in equation (16), and its dual form in Theorem 1. The overcomplete exponential representation given in equation (3) applies to a particular type of hypergraph — namely, a graph with pairwise maximal cliques. Generalizing this type of representation to arbitrary hypergraphs is straightforward. In particular, given the random vector $\mathbf{x}_h = \{x_s \mid s \in h\}$ associated with a particular hyperedge h, we let $J_h = \{j_s \mid j_s \in \mathcal{X}_s\}$ be a multi-index of possible assignments for the subvector $\mathbf{x}_h$. We augment the exponential representation (3) by including the higher order indicator functions $\delta_{h;J}(\mathbf{x}_h) = \prod_{s \in h} \delta_{s;j_s}(x_s)$. For instance, to include a hyperedge (stu) of size three, we include indicators of the form

$$\delta_{stu;jkl}(x_s, x_t, x_u) = \delta_{s;j}(x_s)\delta_{t;k}(x_t)\delta_{u;l}(x_u),$$

corresponding to the function that is equal to one if $(x_s, x_t, x_u) = (j, k, l)$, and zero otherwise.

For a given hypergraph G_{HYP}, let M be the size of its largest hyperedge(s). We define a partition of the hyperedge set E into the disjoint union $\cup_{z=1}^{M} E_z$, where E_z denotes the hyperedges of size z. The full index set $\mathcal{I}$ consists of the following union:

$$\mathcal{I} \quad = \quad \cup_{z=1}^{M} \{(h; J) \mid h \in E_z, J \in \mathcal{X}_h\}.$$

We then consider the exponential family defined by the collection of indicator functions $\{\delta_\alpha \mid \alpha \in \mathcal{I}\}$. As before, we use $\bar{\theta} = \{\bar{\theta}_\alpha \mid \alpha \in \mathcal{I}\}$ to denote the d-dimensional ($d = |\mathcal{I}|$) exponential parameter corresponding to the target distribution $p(\mathbf{x}; \bar{\theta})$. In order to upper bound the log partition function $\Phi(\bar{\theta})$, we make use of the set of all hypertrees contained within G_{HYP}. For a particular hypertree T with associated hyperedge set $E(T)$, let $\theta(T)$ be a hypertree-structured exponential parameter. More formally, letting $\mathcal{I}(T)$ denote the subset of indices corresponding to the hypertree, we require that $\theta(T)$ belongs to the affine subspace $\mathcal{E}(T)$, defined in an analogous manner to equation (8). The quantity $\boldsymbol{\theta} = \{\theta(T)\}$ denotes the full collection of hypertree exponential parameters, which must belong the constraint set $\mathcal{E} = \{\boldsymbol{\theta} \mid \Theta(T) \in \mathcal{E}(T) \; \forall \; T\}$. Finally, we let $\vec{\boldsymbol{\rho}}$ be a probability distribution over these hypertrees. The set $\mathcal{A}(\bar{\theta})$ of feasible pairs $(\boldsymbol{\theta}; \vec{\boldsymbol{\rho}})$ is defined as in equation (15).

With these definitions and Jensen's inequality, we have for any feasible pair $(\boldsymbol{\theta}; \vec{\boldsymbol{\rho}})$ the familiar upper bound $\Phi(\bar{\theta}) \leq \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi(\theta(T))]$. Once again, the natural goal is to optimize the choice of hypertree exponential parameter vectors $\boldsymbol{\theta}$, as well as the choice of distribution $\vec{\boldsymbol{\rho}}$, so as to obtain the tightest possible bound. As we describe in the following section, the former problem of optimizing $\boldsymbol{\theta}$ remains tractable via Lagrangian duality, albeit with a cost that increases exponentially with the treewidth k of the hypertrees. In contrast, the latter problem of optimizing $\vec{\boldsymbol{\rho}}$ is substantially more difficult than the case of ordinary trees ($k = 1$).

E. Dual form of upper bounds

We are now equipped to state the generalization of Theorem 1 to hypertrees. In particular, as with the tree-based bounds, the dual form of the hypertree-based bounds depends on a collection $\tau = \{\tau_h \mid h \in E\}$ of pseudomarginals defined on the hyperedges of the hypergraph G_{HYP}. As with the earlier pseudomarginals $\{\tau_s, \tau_{st}\}$, we require that the hypergraph pseudomarginals are appropriately normalized (i.e., $\sum_{\mathbf{x}'_h} \tau_h(\mathbf{x}'_h) = 1$). In addition, they must satisfy marginalization conditions. In particular, for any nested pair of hyperedge $g < h$, we define the marginal $\tau_{g<h}$ induced by τ_h on g as follows:

$$\tau_{g<h}(\mathbf{x}_g) \quad = \sum_{\{\; \mathbf{x}'_h \;\mid\; \mathbf{x}'_g = \mathbf{x}_g \;\}} \tau_h(\mathbf{x}'_h) \qquad (45)$$

With this notation the relevant constraint set for the pseudomarginals τ, which we denote by $\mathrm{LOCAL}_k(G_{\mathrm{HYP}})$, is defined by the constraints

$$\Big\{ \sum_{\mathbf{x}'_h} \tau_h(\mathbf{x}'_h) = 1 \;\; \forall \; h, \;\; \tau_{g<h}(\mathbf{x}_g) = \tau_g(\mathbf{x}_g) \;\;\; \forall \;\; g < h \;\Big\}, \qquad (46)$$

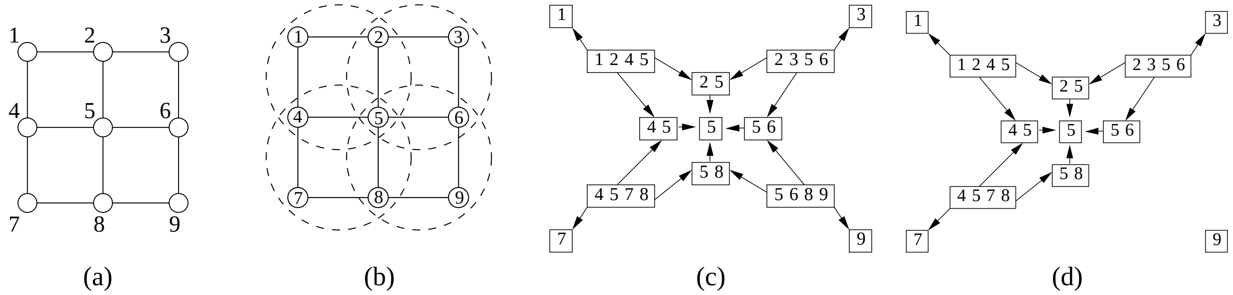
where $k = M - 1$ is one less than the size of the largest hyperedge in G_{HYP}. Note that this constraint set is the natural hypertree-based generalization[8] of the previously defined local constraint set. Moreover, it follows from the junction tree theorem [13], [24] that whenever G_{HYP} is a hypertree (of width k), then $\mathrm{LOCAL}_k(G_{\mathrm{HYP}})$ is a complete description of valid marginal distributions over its hyperedges.

Given a pseudomarginal vector $\tau \in \mathrm{LOCAL}_k(G_{\mathrm{HYP}})$ and a hypertree T, we let $\Pi^T(\tau)$ denote the subcollection $\{\; \tau_h \mid h \in E(T) \;\}$ of pseudomarginals corresponding to hyperedges in the hypertree. This subcollection specifies a unique hypertree-structured distribution via equation (44), which can be used to define the (negative) entropy $\Phi^*(\Pi^T(\tau))$. With this notation, we have the following generalized upper bound on the log partition function:

$$\Phi(\bar{\theta}) \quad \leq \quad \max_{\tau \in \mathrm{LOCAL}_k(G_{\mathrm{HYP}})} \big\{ \langle \tau, \, \bar{\theta} \rangle - \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\tau))] \big\} \qquad (47)$$

In analogy to the tree case (see equation (31)), the inner expectation $\mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\tau))]$ can be explicitly computed, leading to a sum of local entropy terms weighted by *hyperedge appearance* probabilities. The vector $\boldsymbol{\rho_h}$ of these edge appearances must belong to a higher order analog of the spanning tree polytope, which amounts to a hypertree polytope [22].

[8] In particular, for an ordinary graph G, the set $\mathrm{LOCAL}(G)$ defined in equation (19) is equivalent to $\mathrm{LOCAL}_1(G)$, since the largest (hyper)edge in an ordinary tree has size two.



(a) (b) (c) (d)

Fig. 15. Kikuchi clustering on the grid. (a) Original 2D grid. (b) Clustering into groups of four. (c) Associated hypergraph with maximal hyperedges of size $M = 4$. (d) One acyclic hypergraph of width three embedded within (c).

F. Illustrative example

To provide some intuition, we derive a simple form of equation (47) for a particular hypergraph and choice of hypertrees. The original graph that we consider is shown in Figure 15(a). We then cluster the nodes into groups of four, as illustrated in panel (b); this particular choice is known as Kikuchi 4-plaque clustering [46]. On this basis, we can define the collection $E = E_4 \cup E_2 \cup E_1$ of hyperedges, where

$$\begin{aligned} E_4 &:= \{\,(1245),(2356),(4578),(5689)\}, \\ E_2 &:= \{(25),(45),(56),(58)\}, \\ E_1 &:= \{(5),(1),(3),(7),(9)\}. \end{aligned}$$

The hypergraph defined by this hyperedge set is shown in panel (c).

We now consider a convex combination of four hypertrees, each obtained by removing one of the 4-hyperedges from the edge set. For instance, shown in panel (d) is one particular acyclic substructure T^1 with hyperedge set $E(T^1) = E \backslash \{(5689)\}$. To precise, the structure T^1 so defined is a spanning hyperforest, since it consists of two connected components (namely, the isolated hyperedge (9) along with the larger hypertree). This choice, as opposed to a spanning hypertree, turns out to be simplify the development to follow.

To specify the associated hypertree factorization, we first compute the form of φ_h for the maximal hyperedges (i.e., of size four). For instance, looking at the $h = (1245)$, we see that hyperedges (25), (45), (5), and (1) are contained within it. Thus, using the definition in equation (43), we write (suppressing the functional dependence on $\mathbf{x}$):

$$\begin{aligned} \varphi_{1245} &= \frac{\tau_{1245}}{\varphi_{25}\ \varphi_{45}\ \varphi_5 \varphi_1} \\ &= \frac{\tau_{1245}}{\frac{\tau_{25}}{\tau_5}\frac{\tau_{45}}{\tau_5}\tau_5\tau_1} \;=\; \frac{\tau_{1245}\ \tau_5}{\tau_{25}\tau_{45}\tau_1}. \end{aligned}$$

In performing this calculation, we have assumed that the pseudomarginals τ are hypertree-consistent, so that there is no need to distinguish, for instance, between $\tau_{(45)<(1245)}$ and τ_{45}.

Proceeding in this fashion leads an overall factorization of the probability distribution $p(\mathbf{x}; \Pi^{T^1}(\tau))$ based on T^1 of the following form:

$$\left[\frac{\tau_{1245}\ \tau_5}{\tau_{25}\tau_{45}\tau_1}\right]\left[\frac{\tau_{2356}\ \tau_5}{\tau_{25}\tau_{56}\tau_3}\right]\left[\frac{\tau_{4578}\ \tau_5}{\tau_{45}\tau_{58}\tau_7}\right] \times \left[\frac{\tau_{25}}{\tau_5}\right]\left[\frac{\tau_{45}}{\tau_5}\right]\left[\frac{\tau_{56}}{\tau_5}\right]\left[\frac{\tau_{58}}{\tau_5}\right]\left[\tau_1\right]\left[\tau_3\right]\left[\tau_5\right]\left[\tau_7\right]\left[\tau_9\right] \qquad (48)$$

Here each term within square brackets corresponds to φ_h for some hyperedge $h \in E(T^1)$; for instance, the first three terms correspond to the three maximal 4-hyperedges in T^1. Since we are assuming that all of the pseudomarginals are locally consistent, the factorization in equation (48) could be simplified substantially by cancelling terms. However, leaving it in its current form allows us to make the connection to Kikuchi approximations explicit.

Now let $\{T^2, T^3, T^4\}$ denote the four other hyperforests analogous to T^1 — that is, each of them obtained by removing one 4-hyperedge from E. Let $E_4 = \{(1245), (2356), (5689), (4578)\}$ denote the set of all 4-hyperedges. We then form the convex combination of (negative) entropies with uniform weight $1/4$ on each T^i: this convex combination $\sum_{i=1}^{4} \frac{1}{4}\Phi^*(\Pi^{T^i}(\tau))$ takes the form

$$\begin{aligned} &\frac{3}{4}\sum_{h\in E_4}\sum_{\mathbf{x}_h}\tau_h(\mathbf{x}_h)\log\varphi_h(\mathbf{x}_h) + \\ &\qquad \sum_{s\in\{2,4,6,8\}}\sum_{\mathbf{x}_{s5}}\tau_{s5}(\mathbf{x}_{s5})\log\frac{\tau_{s5}(\mathbf{x}_{s5})}{\tau_5(x_5)} \\ &\qquad + \sum_{s\in\{1,3,5,7,9\}}\sum_{x_s}\tau_s(x_s)\log\tau_s(x_s). \qquad (49) \end{aligned}$$

Here the term $3/4$ occurs because each of the 4-hyperedges $h \in E_4$ appears in three of the four hypertrees. All of the (non-maximal) hyperedge terms receive a weight of one, because they appear in all four hypertrees. Overall, then, these weights represent hyperedge appearance probabilities for this particular example, in analogy to ordinary edge appearance probabilities in the tree case. We now simplify the expression in equation (49) by expanding and collecting terms; doing so yields that $-\sum_{i=1}^{4}\frac{1}{4}\Phi^*(\Pi^{T^i}(\tau))$ is equal to the following weighted combination of entropies:

$$\frac{3}{4}\Big[H_{1245}+H_{2356}+H_{5689}+H_{4578}\Big]- \frac{1}{2}\Big[H_{25}+H_{45}+H_{56}+H_{58}\Big] + \frac{1}{4}\Big[H_{1}+H_{3}+H_{7}+H_{9}\Big]. \tag{50}$$

If, on the other hand, starting from equation (49) again, suppose that we included each maximal hyperedge with a weight of 1, instead of $3/4$. Then, after some simplification, we would find that the (negative of the) equation (49) is equal to the following combination of local entropies:

$$H_{1245}+H_{2356}+H_{5689}+H_{4578}- \Big[H_{25}+H_{45}+H_{56}+H_{58}\Big]+H_{5}.$$

This expression is equivalent to the Kikuchi approximation associated with this particular clustering [46]. However, the choice of all ones for the hyperedge appearance probabilities is *invalid* — that is, it could never arise from taking a convex combination of hypertree entropies.

More generally, any entropy approximation formed by taking such convex combinations of hypertree entropies will necessarily be convex. Thus, these functions can be viewed as "convexified" versions of Kikuchi and other free energies. In contrast, with the exception of certain special cases [see, e.g., 26], [29], Kikuchi and other hypergraph-based entropy approximations are typically not convex.

G. *Methods for optimization and open questions*

In this section, we discuss optimization methods for the hypertree-based upper bounds, as well as some associated open questions that arise. We begin by observing that the function (47) to be optimized consists of a convex combination of hypertree entropies (plus a linear term), and is therefore concave in τ. Given this convexity, the optimal τ^* can be found either by standard methods from nonlinear programming [6], or by developing hypertree-reweighted analogs of generalized belief propagation [46]. In parallel to the analog between ordinary sum-product and tree-reweighted sum-product, the computational complexity of these hypertree-weighted algorithms is identical to that of the corresponding version of generalized belief propagation.

For any set of hypertrees that cover the graph, computing the vector of $\boldsymbol{\rho_h}$ of hyperedge appearance probabilities requires computational effort linear in the number of hypertrees. Thus, it is feasible to choose a particular small set of hypertrees that cover the graph, formulate the corresponding optimization problem (47), and then solve it. However, in contrast to the case of spanning trees, *optimizing* the vector $\boldsymbol{\rho_h}$ over all hypertrees is no longer a straight-forward problem. Unlike the case of trees (i.e., hypertrees of width 1), computing the maximum weight hypertree is an NP-hard problem for width two or larger [see 22]. Consequently, we cannot optimize the choice of $\boldsymbol{\rho_h}$ via the conditional gradient method, since it is no longer feasible to solve the linear program that specifies the descent direction. One open research question, then, is to develop methods for approximately solving the maximum weight hypertree problem, and to apply them to tightening the hypertree bounds described here. In addition, just as with ordinary cluster variational approximations, there are various open questions associated with which clusters to choose, and how to assess their effect on the approximation accuracy [46], [47], [41], [21]. A desirable feature of the convex framework presented here is that using more complex hypertrees is always guaranteed to produce tighter upper bounds, since the optimization problems are naturally nested. Specifically, Wiegerinck [43] has compared hypertree-reweighted forms of generalized belief propagation (GBP) to ordinary generalized BP, as well as the standard and tree-reweighted BP algorithms. Interestingly, his results show that using hypertree-reweighted forms of GBP not only leads to tighter bounds (which is theoretically guaranteed), but also seems to lead to consistently better approximations to the marginals. In contrast, there are certain cases in which using ordinary Kikuchi approximations may lead to worse results that the usual Bethe approximation [21], [43] (but see Yedidia et al. [47] for the relevance of "max-ent normal" approximations). A final interesting direction for future research is to explore the effect of different choices of hypertrees on the accuracy of the pseudomarginal approximations.

VII. Conclusions

In this paper, we have developed and analyzed a new class of upper bounds for the log partition function of an arbitrary Markov random field. The basic form of these upper bounds follows by forming a mixture in the exponential domain of tractable (e.g., tree-structured) distributions, and applying Jensen's inequality. Using convex duality, we showed that the optimal form of such bounds can be obtained by solving a convex variational problem. We explored in detail the case of spanning trees, and showed how the Lagrangian reformulation allows us to optimize efficiently — though implicitly — over all possible choices of exponential parameters for the tractable distributions, as well as over all choices of weights for the exponential mixture. The cost function in the spanning tree case, while similar to the Bethe variational problem [46], is distinguished by its convexity, which holds for an arbitrary graph. This derivation provides a novel perspective on the Bethe variational problem. In addition, we established a concrete link to the sum-product algorithm by deriving a tree-reweighted version of the sum-product updates for solving our convex program. More generally, we discussed how stronger bounds can be obtained by taking convex combinations of hypertrees, and the resulting link to Kikuchi and other entropy variational problems [e.g., 46], [47], [27], [29], [26], [37]. This extension raises a number of open questions, including how to choose the base set of hypertrees, as well as how to optimize (at least approximately) the choices of hyperedge weights.

We conclude by discussing a few directions for future research. First, the basic idea of generating upper bounds using convex combinations need not be restricted to trees or hypertrees. One interesting direction is developing upper bounds based on other types of convex combinations, or combinations that involved distributions over parameters as well as weights. Second, in previous work [37], [35], we have derived lower and upper bounds on the error in the sum-product algorithm (i.e., the difference between the exact marginals and the approximate marginals computed by sum-product). In conjunction with techniques described in this paper, it is possible to efficiently compute bounds on the approximation error of the sum-product algorithms, as well as various generalizations thereof [e.g., 46]. It remains to explore the usefulness of these bounds for larger scale practical problems. Lastly, the optimization problems defined in this paper define a new convex function that can be viewed as a surrogate to the log partition function. This surrogate has a number of possible applications to parameter estimation, as our initial work [38] in this direction has suggested.

Acknowledgements: This work was supported by ODDR&E MURI Grant DAAD19-00-1-0466 through ARO; by ONR Grant N00014-00-1-0089; and by AFOSR Grant F49620-00-1-0362. The authors wish to express their thanks to Yee Whye Teh and Max Welling for providing code for performing exact inference on grids; to David Karger for providing pointers to the spanning tree polytope; to Nati Srebro for helpful discussions on hypertrees; and to the anonymous referees for their comments and constructive criticism that helped to improve the presentation of the work.

APPENDIX

A. Additional remarks on Theorem 1

In this section, we establish the validity of the bound in equation (27) for arbitrary $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, which enables us to remove the restriction that $\rho_e > 0$, and instead enforce the weaker constraint $\rho_e \geq 0$. Recall the definition of $\mathrm{MARG}(\boldsymbol{\phi})$ from equation (6). In the particular case of the overcomplete exponential parameterization of equation (3) for a pairwise Markov random field based on a graph G, $\mathrm{MARG}(\boldsymbol{\phi})$ consists the set of all single node and edgewise marginals $\mu = \{\mu_s, \mu_{st}\}$ that arise from taking expectations with respect to some distribution $p(\mathbf{x}; \theta)$. We denote this set of realizable pairwise and singleton marginals associated with the graph G by $\mathrm{MARG}(G)$.

By the conjugate duality between the log partition function Φ and the negative entropy Φ^*, we are guaranteed the following variational representation of the log partition function:

$$\Phi(\bar{\theta}) = \max_{\mu \in \mathrm{MARG}(G)} \big\{ \langle \mu, \, \bar{\theta} \rangle - \Phi^*(\mu) \big\}. \tag{51}$$

Our strategy is to exploit this variational representation in order to obtain an upper bound on $\Phi(\bar{\theta})$.

The first requirement is a lower bound on $\Phi^*(\mu)$, or equivalently an upper bound on the entropy $-\Phi^*(\mu)$. For any spanning tree T of the graph G, consider the tree-structured distribution $p(\mathbf{x}; \Pi^T(\mu))$ defined as in equation (21). We first claim that the inequality

$$-\Phi^*(\mu) \quad \leq \quad -\Phi^*(\Pi^T(\mu)) \tag{52}$$

holds for any tree $T \in \mathfrak{T}$, meaning that the entropy of the moment-matched tree-structured distribution must be at least as large as the entropy of the distribution on the full graph with cycles. A full proof of this inequality can be found in Wainwright and Jordan [39]. The basic intuition follows from the interpretation of exponential family members as maximum entropy distributions subject to constraints [12]. The distribution on the full graph with cycles arises from a maximization problem with *more* constraints, and hence it must have a lower entropy than the tree-structured version.

Since inequality (52) holds for any spanning tree T, so that we can take a convex combination of such bounds using $\vec{\boldsymbol{\rho}} = \{\rho(T)\}$ as non-negative weights, which yields:

$$\Phi^*(\mu) \;\geq\; \sum_{T} \rho(T) \Phi^*(\Pi^T(\mu)) \;=\; \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\mu))]. \tag{53}$$

The second requirement is an outer bound on the marginal polytope $\mathrm{MARG}(G)$. It is clear that any single node marginal μ_s must be normalized (i.e., $\sum_{j \in \mathcal{X}_s} \mu_{s;j} = 1$), and moreover, that any joint marginal

μ_{st} on the edge $(s,t) \in E$ must marginalize appropriately (i.e., $\sum_{j \in \mathcal{X}_s} \mu_{st;jk} = \mu_{t;k}$). Therefore, the constraint set $\mathrm{LOCAL}(G)$ defined in equation (19) is an outer bound on the marginal polytope (i.e., $\mathrm{MARG}(G) \subseteq \mathrm{LOCAL}(G)$).

Finally, to obtain the desired upper bound, we first apply equation (53) to the variational formulation of equation (51), thereby obtaining the upper bound:

$$\Phi(\bar{\theta}) \quad \leq \quad \max_{\mu \in \mathrm{MARG}(G)} \big\{ \langle \mu, \, \bar{\theta} \rangle - \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\mu))] \big\}.$$

We complete the proof by using the fact that $\mathrm{MARG}(G) \subseteq \mathrm{LOCAL}(G)$ to write:

$$\Phi(\bar{\theta}) \quad \leq \quad \max_{\tau \in \mathrm{LOCAL}(G)} \big\{ \langle \tau, \, \bar{\theta} \rangle - \mathbb{E}_{\vec{\boldsymbol{\rho}}}[\Phi^*(\Pi^T(\tau))] \big\}.$$

Thus, the bound of equation (27) holds for any choice of $\vec{\boldsymbol{\rho}}$, and hence any $\boldsymbol{\rho_e} \in \mathbb{T}(G)$.

B. *Proof of Theorem 2:*

We first prove the following lemma:

Lemma 2. *For each valid $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, the function R is differentiable in $\boldsymbol{\rho_e}$, with partial derivatives given by (negative) mutual information terms:*

$$\frac{\partial R}{\partial \rho_{st}}(\boldsymbol{\rho_e}; \bar{\theta}) \quad = \quad -I_{st}(\tau^*_{st}(\boldsymbol{\rho_e})). \tag{54}$$

Proof: Consider the function $\varphi_{\bar{\theta}}(\tau; \boldsymbol{\rho_e}) = \langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})$. It is continuous in τ and $\boldsymbol{\rho_e}$, and for each fixed τ, $\varphi_{\bar{\theta}}(\tau; \cdot)$ is linear and hence convex and differentiable in $\boldsymbol{\rho_e}$. Moreover, the constraint set $\mathrm{LOCAL}(G)$ is compact, and by Theorem 1, for each fixed valid $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, the optimum that defines R — namely, $\max_{\tau \in \mathrm{LOCAL}(G)} \varphi_{\bar{\theta}}(\tau; \boldsymbol{\rho_e})$ —– is attained at a unique point τ^*. Therefore, by results on the suprema of convex functions [e.g., 16], the function R is differentiable at $\boldsymbol{\rho_e}$, with partial derivatives given by

$$\begin{aligned} \frac{\partial R}{\partial \rho_{st}}(\boldsymbol{\rho_e}; \bar{\theta}) \quad &= \quad \frac{\partial}{\partial \rho_{st}} \big\{ \langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e}) \big\} \Big|_{\tau^*(\boldsymbol{\rho_e})} \\ &= \quad -I_{st}(\tau^*_{st}(\boldsymbol{\rho_e})). \end{aligned}$$

Proof of Theorem 2:

(a) The bound of equation (27) holds for all $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, from which equation (34) follows. Observe that $\langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})$ is linear in $\boldsymbol{\rho_e}$. Therefore, $R(\boldsymbol{\rho_e}; \bar{\theta})$ is the maximum over a collection of linear functions, and so is convex [6] as a function of $\boldsymbol{\rho_e}$. From Theorem 1, for any fixed $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, the value $R(\boldsymbol{\rho_e}; \bar{\theta})$ gives an upper bound on $\Phi(\bar{\theta})$. Therefore, minimizing over all $\boldsymbol{\rho_e}$ yields the optimal value of the upper bound, as in equation (34).

(b) For each spanning tree T and each edge $(s,t) \in E$, let $\delta[(s,t) \in E]$ be the indicator function for the event $(s,t) \in T$. Now R is convex and the constraint set $\mathbb{T}(G)$ is linear. Therefore, by standard necessary conditions for the minimum of a convex function over a convex set [6], we are guaranteed that for all $\boldsymbol{\rho_e} = \{\rho_{st}\} \in \mathbb{T}(G)$:

$$\begin{aligned} \sum_{(s,t) \in E} \frac{\partial R}{\partial \rho_{st}}(\boldsymbol{\rho_e}^*; \bar{\theta}) \, \big\{ \rho_{st} - \rho^*_{st} \big\} \; &= \\ \sum_{(s,t) \in E} I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*)) \, \big\{ \rho_{st} - \rho^*_{st} \big\} \; &\geq \; 0. \end{aligned} \tag{55}$$

Here we have used Lemma 2 in order to compute the partial derivatives of R in equation (55).

For a given spanning tree T, consider the indicator $\delta[e \in T]$, a vector of length $|E|$ with the e^{th} element equal to one if e belongs to T, and zero otherwise. The vector so defined is an element of the spanning tree polytope (an extreme point, in fact), so that equation (55) applies, in particular, to it.

Now, since $\boldsymbol{\rho_e}^*$ is a member of the spanning tree polytope, there must exist a distribution $\vec{\boldsymbol{\rho}}^*$ over spanning trees such that for each $(s,t) \in E$, we have the relation $\sum_{T \in \mathfrak{T}} \rho^*(T) \; \delta[(s,t) \in T] = \rho^*_{st}$, or equivalently, such that:

$$0 \quad = \quad \sum_{T \in \mathfrak{T}} \rho^*(T) \, \big\{ \delta[(s,t) \in T] - \rho^*_{st} \big\}. \tag{56}$$

We multiply equation (56) by $I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*))$, and sum the resultant collection of equations (one for each $(s,t) \in E$) to obtain:

$$\begin{aligned} 0 \quad &= \quad \sum_{(s,t) \in E} I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*)) \sum_{T \in \mathfrak{T}} \rho^*(T) \, \big\{ \delta[(s,t) \in T] - \rho^*_{st} \big\} \\ &= \quad \sum_{T \in \mathfrak{T}} \rho^*(T) \Big[\sum_{(s,t) \in E} I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*)) \, \big\{ \delta[(s,t) \in T] - \rho^*_{st} \big\} \Big] \end{aligned} \tag{57}$$

Now by equation (55), for each T, the term within square brackets on the RHS of equation (57) is non-positive. Since the overall sum is equal to zero, we must have that

$$\sum_{(s,t) \in E} I_{st}(\tau^*_{st}(\boldsymbol{\rho_e}^*)) \, \big\{ \delta[(s,t) \in T] - \rho^*_{st} \big\} \; = \; 0.$$

for all $T \in \mathrm{supp}(\vec{\boldsymbol{\rho}}^*)$. We thus have established equation (35).

(c) The function $\langle \tau, \, \bar{\theta} \rangle - Q(\tau; \boldsymbol{\rho_e})$ is continuous as a function of both τ and $\boldsymbol{\rho_e}$. By Theorem 1, it is concave in τ; moreover, it is linear and hence convex in $\boldsymbol{\rho_e}$. Moreover, the constraint sets $\mathrm{LOCAL}(G)$ and $\mathbb{T}(G)$ are both convex and compact. Equation (36) therefore follows from standard minimax results [16]. □

C. *Tree-reweighted sum-product*

In this appendix, we establish that Algorithm 1 solves the optimization problem (38). We begin by observing that any set of messages $\mathbf{M}$ can be used to define a set of singleton pseudomarginals

$$\tau_s(x_s) \;=\; \kappa\, \exp\big(\bar{\theta}_s(x_s)\big) \prod_{v\in\Gamma(s)} \big[M_{vs}(x_s)\big]^{\rho_{vs}}, \tag{58}$$

as well as a set of joint pseudomarginals

$$\tau_{st}(x_s,x_t) \propto \varphi_{st}(x_s,x_t;\bar{\theta})\; \frac{\prod_{v\in\Gamma(s)\backslash t}\big[M_{vs}(x_s)\big]^{\rho_{vs}}}{\big[M_{ts}(x_s)\big]^{(1-\rho_{st})}}\; \frac{\prod_{v\in\Gamma(t)\backslash s}\big[M_{vt}(x_t)\big]^{\rho_{vt}}}{\big[M_{st}(x_t)\big]^{(1-\rho_{ts})}}, \tag{59}$$

where the notation $\varphi_{st}(x_s,x_t;\bar{\theta})$ is short-hand for

$$\varphi_{st}(x_s,x_t;\bar{\theta}) \;=\; \exp\Big(\frac{\bar{\theta}_{st}(x_s,x_t)}{\rho_{st}} + \bar{\theta}_s(x_s) + \bar{\theta}_t(x_t)\Big).$$

Moreover, the term κ denotes a constant chosen so as to ensure that the normalization conditions (e.g., $\sum_{x'_s}\tau_s(x'_s) = 1$) are satisfied.

Proposition 1. *For any valid $\boldsymbol{\rho_e} \in \mathbb{T}(G)$, the pseudo-marginals τ^* specified by a fixed point $\mathbf{M}^*$ of Algorithm 1 via equations* (58) *and* (59) *attain the global optimum of the variational problem of equation* (27).

Proof: As with the work of Yedidia et al. [46] on ordinary sum-product and the Bethe problem, we show that any fixed point of Algorithm 1 satisfies the conditions to be a stationary point of the Lagrangian associated with the constrained optimization problem (27). In contrast to the ordinary Bethe problem, our problem is convex (by Theorem 1), so that these stationarity conditions are sufficient to ensure that we have found a global optimum. For each edge (s,t), let $\lambda_{ts}(x_s)$ be a Lagrange multiplier associated with the marginalization constraint $C_{ts}(x_s) := \sum_{x_t}\tau_{st}(x_s,x_t) - \tau_s(x_s) = 0$. We then form the Lagrangian associated with the variational problem of equation (27) (where we enforce the normalization and non-negativity constraints explicitly without multipliers):

$$\begin{aligned}\mathcal{L}(\tau;\lambda) = \langle\tau,\,\bar{\theta}\rangle - Q(\tau;\boldsymbol{\rho_e}) + \\ \sum_{(s,t)\in E}\Big\{\lambda_{ts}(x_s)C_{ts}(x_s) + \lambda_{st}(x_t)C_{st}(x_t)\Big\}\end{aligned} \tag{60}$$

Taking the derivative with respect to $\tau_s(x_s)$ and setting it to zero yields

$$\log\tau_s(x_s) = C + \phi_s(x_s;\bar{\theta}_s) + \sum_{u\in\Gamma(s)}\lambda_{us}(x_s). \tag{61}$$

Similarly, taking derivatives with respect to $\tau_{st}(x_s,x_t)$ yields

$$\begin{aligned}\rho_{st}\log\frac{\tau_{st}(x_s,x_t)}{\big(\sum_{x_s}\tau_{st}(x_s,x_t)\big)\big(\sum_{x_t}\tau_{st}(x_s,x_t)\big)} = \\ C + \phi_{st}(x_s,x_t;\bar{\theta}_{st}) - \lambda_{st}(x_t) - \lambda_{ts}(x_s).\end{aligned} \tag{62}$$

In these equations, C denotes a constant (independent of $\mathbf{x}$) that will be chosen to ensure that the normalization conditions are satisfied. Imposing the fact that an optimal solution must satisfy the marginalization conditions and then using equation (61) allows us to re-arrange equation (62) into the form:

$$\begin{aligned}\log\tau_{st}(x_s,x_t) = C + \frac{\bar{\theta}_{st}(x_s,x_t)}{\rho_{st}} + \bar{\theta}_s(x_s) + \bar{\theta}_t(x_t) \\ + \sum_{u\in\Gamma(s)}\lambda_{us}(x_s) + \sum_{u\in\Gamma(t)}\lambda_{ut}(x_t) - \frac{\lambda_{ts}(x_s)}{\rho_{st}} - \frac{\lambda_{st}(x_t)}{\rho_{st}}\end{aligned} \tag{63}$$

We now define (log) "messages" via $\log M_{st}(x_t) := \frac{1}{\rho_{st}}\lambda_{st}(x_t)$. Making these replacements in equation (61) and equation (63) yields the expressions for τ_s and τ_{st} in equations (58) and (59) respectively.

Finally, we need to ensure that τ_s and τ_{st} belong to the constraint set $\mathrm{LOCAL}(G)$. Since we imposed the normalization constraints explicitly, we need to update the "messages" so that the marginalization constraint $\sum_{x_s}\tau_{st}(x_s,x_t) = \tau_s(x_s)$ is satisfied. Enforcing this constraint yields the message update in equation (39).

References

[1] S. Amari. Differential geometry of curved exponential families — curvatures and information loss. *Annals of Statistics*, 10(2):357–385, 1982.

[2] S. Amari. Information geometry on a hierarchy of probability distributions. *IEEE Trans. on Information Theory*, 47(5):1701–1711, 2001.

[3] O. E. Barndorff-Nielson. *Information and exponential families*. Wiley, Chichester, 1978.

[4] R. J. Baxter. *Exactly solved models in statistical mechanics*. Academic Press, New York, 1982.

[5] C. Berge. *Hypergraphs*. North-Holland Publishing Company, Amsterdam, 1989.

[6] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1995.

[7] N. Biggs. *Algebraic graph theory*. Cambridge University Press, Cambridge, 1993.

[8] B. Bollobás. *Modern graph theory*. Springer-Verlag, New York, 1998.

[9] L.D. Brown. *Fundamentals of statistical exponential families*. Institute of Mathematical Statistics, Hayward, CA, 1986.

[10] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.

[11] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Info. Theory*, IT-14:462–467, 1968.

[12] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

[13] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag, 1999.

[14] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[15] T. Heskes, K. Albers, and B. Kappen. Approximate inference and constrained optimization. In *Uncertainty in Artificial Intelligence*, volume 13, page to appear, 2003.

[16] J. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms*, volume 1. Springer-Verlag, New York, 1993.

[17] T. S. Jaakkola and M. Jordan. Recursive algorithms for approximating probabilities in graphical models. In *NIPS 9*, 1996.

[18] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal Comput.*, 22:1087–1116, 1993.

[19] M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. MIT Press, 1999.

[20] D. Jungnickel. *Graphs, networks, and algorithms*. Springer, New York, 1999.

[21] H. J. Kappen and W. Wiegerinck. Novel iteration schemes for the cluster variation method. In *Neural Information Processing Systems 14*, pages 415—422. MIT Press, Cambridge, Ma, 2002.

[22] D. Karger and N. Srebro. Learning Markov networks: maximum bounded tree-width graphs. In *Symposium on Discrete Algorithms*, pages 392–401, 2001.

[23] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, 47:498–519, February 2001.

[24] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[25] M.A.R. Leisink and H.J. Kappen. A tighter bound for graphical models. In *NIPS 13*, pages 266–272. MIT Press, 2001.

[26] R. J. McEliece and M. Yildirim. Belief propagation on partially ordered sets. In D. Gilliam and J. Rosenthal, editors, *Mathematical Theory of Systems and Networks*. Institute for Mathematics and its Applications, 2002.

[27] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, January 2001.

[28] L. Onsager. Crystal statistics I: A two-dimensional model with an order-disorder transition. *Physical Review*, 65:117–149, 1944.

[29] P. Pakzad and V. Anantharam. Iterative algorithms and free energy minimization. In *CISS*, March 2002.

[30] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufman, San Mateo, 1988.

[31] G. Potamianos and J. Goutsias. Stochastic approximation algorithms for partition function estimation of Gibbs random fields. *IEEE Trans. Info. Theory*, 43(6):1948–1965, November 1997.

[32] G. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[33] R. P. Stanley. *Enumerative combinatorics*, volume 1. Cambridge University Press, Cambridge, UK, 1997.

[34] J. H. van Lint and R. M. Wilson. *A course in combinatorics*. Cambridge University Press, Cambridge, 1992.

[35] M. J. Wainwright. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, MIT, January 2002.

[36] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. In *Uncertainty in Artificial Intelligence*, volume 18, pages 536–543, August 2002.

[37] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Info. Theory*, 49(5):1120–1146, 2003.

[38] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching. In *Workshop on Artificial Intelligence and Statistics*, January 2003.

[39] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical report, UC Berkeley, Department of Statistics, No. 649, September 2003.

[40] M. J. Wainwright and M. I. Jordan. Variational inference in graphical models: The view from the marginal polytope. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, October 2003.

[41] M. Welling. On the choice of regions for generalized belief propagation. In *Uncertainty in Artificial Intelligence*, 2004.

[42] D. J. A. Welsh. *Matroid theory*. Academic Press, New York, 1976.

[43] W. Wiegerinck. Approximations with reweighted generalized belief propagation. In *Workshop on Artificial Intelligence and Statistics*, January 2005.

[44] W. Wiegerinck and T. Heskes. Fractional belief propagation. In *NIPS*, volume 12, page to appear, 2002.

[45] A. S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, 2002.

[46] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR2001-22, Mitsubishi Electric Research Labs, January 2002.

[47] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR2001-40, Mitsubishi Electric Research Labs, May 2004.

[48] J. Zhang. The application of the Gibbs-Bogoliubov-Feynman inequality in mean-field calculations for Markov random-fields. *IEEE Trans. on Image Processing*, 5(7):1208–1214, July 1996.