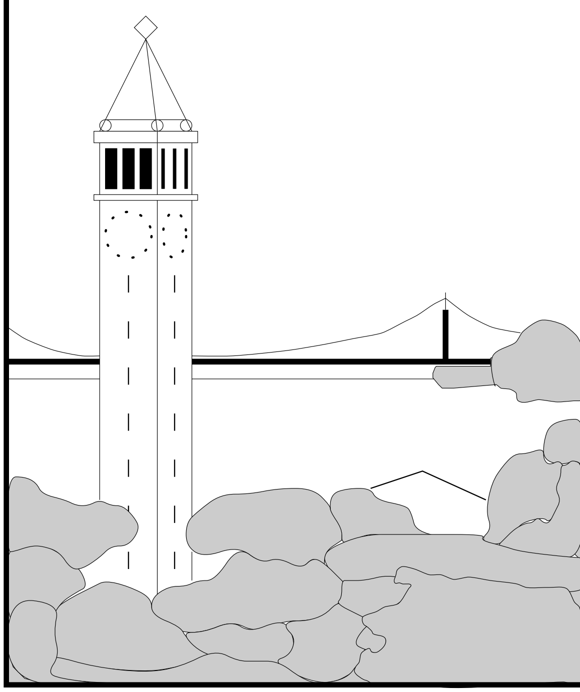
MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches

Martin J. Wainwright, Tommi S. Jaakkola and Alan S. Willsky



Report No. UCB/CSD-3-1269

August 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches

Martin J. Wainwright[*] wainwrig@eecs.berkeley.edu
Tommi S. Jaakkola[†] tommi@ai.mit.edu
Alan S. Willsky[†] willsky@mit.edu

Department of EECS[*]
UC Berkeley, CA 94720

Department of EECS[†]
MIT, Cambridge, MA 02139

August 29, 2003

Abstract

We develop and analyze methods for computing provably optimal *maximum a posteriori* (MAP) configurations for a subclass of Markov random fields defined on graphs with cycles. By decomposing the original distribution into a convex combination of tree-structured distributions, we obtain an upper bound on the optimal value of the original problem (i.e., the log probability of the MAP assignment) in terms of the combined optimal values of the tree problems. We prove that this upper bound is met with equality if and only if the tree distributions share an optimal configuration in common. An important implication is that any such shared configuration must also be a MAP configuration for the original distribution. Next we develop two approaches to attempting to obtain tight upper bounds: (a) a *tree-relaxed linear program* (LP), which is derived from the Lagrangian dual of the upper bounds; and (b) a *tree-reweighted message-passing algorithm* that is related to but distinct from the max-product (min-sum) algorithm. Finally, we discuss the conditions that govern when the relaxation is tight, in which case the MAP configuration can be obtained. The analysis described here generalizes naturally to convex combinations of hypertree-structured distributions.

Keywords: Markov random fields; max-product algorithm; min-sum algorithm; message-passing algorithms; approximate inference; MAP estimation; iterative decoding; integer programming; linear programming relaxation; marginal polytope.

1 Introduction

Integer programming problems arise in various fields, including communication theory, error-correcting coding, image processing, statistical physics and machine learning [e.g., 29, 32, 8]. Many such problems can be formulated in terms of Markov random fields [e.g., 8, 14], in which the cost function corresponds to a graph-structured probability distribution, and the goal is to find the *maximum a posteriori* (MAP) configuration. It is well-known that the complexity of solving the MAP estimation problem on a Markov random field (MRF) depends critically on the structure of the underlying graph. For cycle-free graphs (also known as trees), the MAP problem can be solved by a form of non-serial dynamic programming known as the max-product or min-sum algorithm [e.g., 2, 14, 15]. This algorithm, which entails passing "messages" from node to node, represents a generalization of the Viterbi algorithm [33] from chains to arbitrary cycle-free graphs. In recent years, the max-product algorithm has also been studied in application to graphs with cycles as a method for computing approximate MAP assignments [e.g., 1, 20, 21, 22, 27, 36]. Although the method may perform well in practice, it is no longer guaranteed to output the correct MAP assignment, and

Work supported in part by ODDR&E MURI Grant DAAD19-00-1-0466 through the ARO; by ONR N00014-00-1-0089; and by the AFOSR F49620-00-1-0161

it is straightforward to demonstrate problems on which it specifies an incorrect (i.e., non-optimal) assignment.

In this paper, we present and analyze new methods for computing MAP configurations for MRFs defined on graphs with cycles. The basic idea is to use a convex combination of tree-structured distributions to derive upper bounds on the cost of a MAP configuration. We prove that any such bound is met with equality if and only if the trees share a common optimizing configuration; moreover, any such shared configuration must be MAP-optimal for the original problem. Consequently, when the bound is tight, obtaining an MAP configuration for a graphical model with cycles — in general, a *very* difficult problem — is reduced to the easy task of examining the optima of a collection of tree-structured distributions.

Accordingly, we focus our attention on the problem of obtaining tight upper bounds, and propose two methods directed to this end. Our first approach is based on the convexity of the upper bounds, and the associated theory of Lagrangian duality. We begin by re-formulating the exact MAP estimation problem on a graph with cycles as a linear program (LP) over the so-called marginal polytope. We then consider the Lagrangian dual of the problem of optimizing our upper bound. In particular, we prove that this dual is another LP, one which has a natural interpretation as a relaxation of the LP for exact MAP estimation. The relaxation is obtained by replacing the marginal polytope for the graph with cycles, which is a very complicated set in general, by an outer bound with simpler structure. This outer bound is an exact characterization of the marginal polytope of any tree-structured distribution, for which reason we refer to this approach as a *tree-based relaxation*.

The second method consists of a class of message-passing algorithms designed to find a collection of tree-structured distributions that share a common optimum. The resulting algorithm, though similar to the standard max-product (or min-sum) algorithm [e.g., 22, 36], differs from it in a number of important ways. In particular, under the so-called optimum specification criterion, fixed points of our *tree-reweighted max-product algorithm* specify a MAP-optimal configuration with a guarantee of correctness. We also prove that under this condition, fixed points of the tree-reweighted max-product updates correspond to dual-optimal solutions of the tree-relaxed linear program. As a corollary, we establish that the ordinary max-product algorithm on trees is solving the dual of an exact LP formulation of the MAP estimation problem.

Overall, this paper establishes connections between two approaches to solving the MAP estimation problem: LP relaxations of integer programming problems [e.g., 7, 16, 25, 31], and (approximate) dynamic programming methods using message-passing in the max-product algebra. This connection has links to the recent work of Yedidia et al. [39], who showed that the sum-product algorithm has a variational interpretation involving the so-called Bethe free energy. In addition, the work described here is linked in spirit to our previous work [35, 34], in which we showed how to upper bound the log partition function using a "convexified form" of the Bethe free energy. Whereas this convex variational problem led to a method for computing approximate marginal distributions, the current paper deals exclusively with the problem of computing MAP configurations. Importantly and in sharp contrast with our previous work, there is a non-trivial set of problems for which the upper bounds of this paper are tight, in which case the MAP-optimal configuration can be obtained by the techniques described here. For instance, as documented in other work, we have applied these techniques successfully to data association problems involving multiple targets and sensors [13], and as a decoding method for turbo-like and LDPC codes [18, 19].

The remainder of this paper is organized as follows. Section 2 provides necessary background on graph theory and graphical models, as well as some preliminary details on marginal polytopes, and a formulation of the MAP estimation problem. In Section 3, we introduce the basic form of the upper bounds on the log probability of the MAP assignment, and then develop necessary and sufficient conditions for these bounds to be met with equality. In Section 4, we first discuss how the MAP integer programming problem has an equivalent formulation as a linear program (LP) over the marginal polytope. We then prove that the Lagrangian dual of the problem of optimizing our upper bounds has a natural interpretation as a tree-relaxation of the original LP. Section 5 is devoted to the development of iterative message-passing algorithms and their relation to the dual of the LP relaxation. We conclude in Section 6 with a discussion and extensions to the analysis presented here.

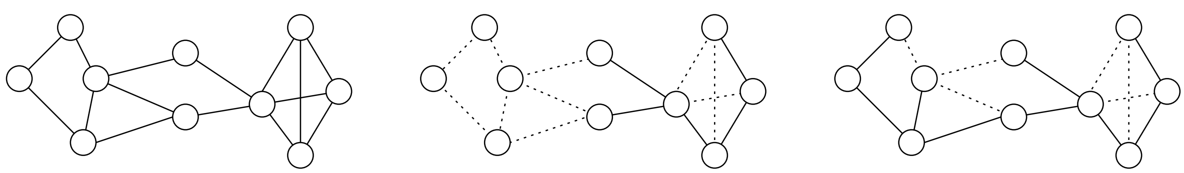
2 Preliminaries

This section provides the background and some preliminary developments necessary for subsequent sections. We begin with a brief overview of some graph-theoretic basics; we refer the reader to the books [9, 10] for additional background on graph theory. We then describe the formalism of Markov random fields; more details can be found in various sources [e.g., 12, 14, 28]. We conclude with a precise formulation of the MAP estimation problem for a Markov random field.

2.1 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. In this paper, we consider only simple graphs, for which multiple edges between the same pair of vertices are forbidden. 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 *clique* of the graph G is a fully-connected subset C of the vertex set (i.e., $(s,t) \in E$ for all $s, t \in C$). The clique C is *maximal* if it is not properly contained within any other clique. 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 subgraph of $G = (V, E)$ is a graph $H = (V(H), E(H))$ where $V(H)$ (respectively $E(H)$) are subsets of V (respectively E). Of particular importance to our analysis are those (sub)graphs without cycles. More precisely, a *tree* is a cycle-free subgraph $T = (V(T), E(T))$; it 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.



(a) (b) (c)

Figure 1. (a) Graph with cycles. (b) A tree is a cycle-free subgraph. (c) A spanning tree reaches every vertex of the graph.

2.2 Markov random fields

A Markov random field (MRF) is defined on the basis of an undirected graph $G = (V, E)$ in the following way. For each $s \in V$, let x_s be a random variable taking values in some sample space

$\mathcal{X}_s$. This paper deals exclusively with the discrete case, for which x_s takes values in the finite alphabet $\mathcal{X}_s := \{0, \ldots, m_s - 1\}$. By concatenating the variables at each node, we obtain a random vector $\mathbf{x} = \{ x_s \mid s \in V\}$ with $n = |V|$ elements. Observe that $\mathbf{x}$ itself takes 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 $A \subseteq V$, 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.

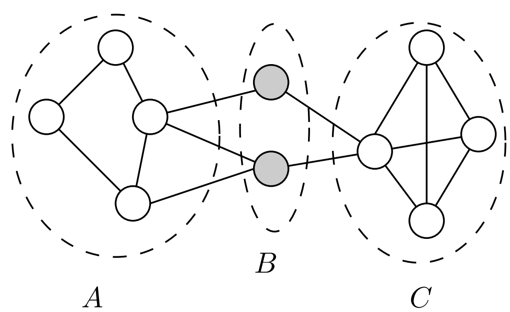


Figure 2. Illustration of the relation between conditional independence and graph separation. Here the set of nodes B separates A and C, so that for a Markov random field, $\mathbf{x}_{A|B}$ and $\mathbf{x}_{C|B}$ must be conditionally independent.

The well-known *Hammersley-Clifford theorem* [e.g., 12, 28] asserts that any Markov random field 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. For the sake of development in the sequel, it is convenient to describe this decomposition in terms of exponential families [e.g., 3, 4]. We begin with some necessary notation. A *potential function* associated with a given clique C is mapping $\phi : \mathcal{X}^n \to \mathbb{R}$ that depends only on the subcollection $\mathbf{x}_C := \{x_s \mid s \in C\}$. 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 $\boldsymbol{\phi}(\mathbf{x})$.

With this set-up, the collection of strictly positive Markov random fields associated with the graph G and potential functions $\boldsymbol{\phi}$ can be represented as the *exponential family* $\{p(\mathbf{x}; \theta) \mid \theta \in \mathbb{R}^d\}$, where

$$p(\mathbf{x}; \theta) \quad \propto \quad \exp\big\{\langle \theta, \boldsymbol{\phi}(\mathbf{x}) \rangle\big\} \;\equiv\; \exp\big\{ \sum_{\alpha \in \mathcal{I}} \theta_\alpha \phi_\alpha(\mathbf{x}) \big\}. \tag{1}$$

Note that each vector $\theta \in \mathbb{R}^d$ indexes a particular Markov random field $p(\mathbf{x}; \theta)$ in this exponential family.

Example 1. The *Ising model* of statistical physics [e.g., 5] provides a simple illustration of a collection of MRFs in this form. This model involves a binary random vector $\mathbf{x} \in \{0, 1\}^n$, with a distribution defined by potential functions only on cliques of size at most two (i.e., vertices and edges). As a result, the exponential family in this case takes the form:

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

Here θ_{st} is the weight on edge (s,t), and θ_s is the parameter for node s. In this case, the index set $\mathcal{I}$ consists of the union $V \cup E$. Note that the set of potentials $\{x_s, s \in V\} \cup \{x_s x_t, (s,t) \in E\}$ is a basis for all multinomials on $\{0,1\}^n$ of maximum degree two that respect the structure of G. $\diamondsuit$

When the collection of potential functions $\boldsymbol{\phi}$ do not satisfy any linear constraints, then the representation (1) is said to be *minimal* [3, 4]. For example, the Ising model (2) is minimal, because there is no linear combination of the potentials $\boldsymbol{\phi} = \{x_s,\ s \in V\} \cup \{x_s x_t,\ (s,t) \in E\}$ that is equal to a constant for all $\mathbf{x} \in \{0,1\}^n$. In contrast, it is often convenient to consider an *overcomplete representation*, in which the potential functions $\boldsymbol{\phi}$ do satisfy linear constraints, and hence are no longer a basis. More specifically, our development in the sequel makes extensive use of an overcomplete representation in which the basic building blocks are *indicator functions* of the form $\delta_j(x_s)$ — the function that is equal to one if $x_s = j$, and zero otherwise. In particular, for a Markov random field with interactions between at most pairs of variables, we use the following collection of potential functions:

$$\{\delta_j(x_s) \ \big| \ j \in \mathcal{X}_s\} \quad \text{for } s \ \in \ V, \tag{3a}$$

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

which we refer to as the *canonical overcomplete representation*. This representation involves a total of $d := \sum_{s\in V} m_s + \sum_{(s,t)\in E} m_s m_t$ potential functions, indexed by the set

$$\mathcal{I} \quad := \quad \big[\cup_{s\in V}\{(s;j), j \in \mathcal{X}_s\}\big] \ \cup \ \big[\cup_{(s,t)\in E}\{(st;jk), (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}\big]. \tag{4}$$

The overcompleteness of the representation is manifest in various linear constraints among the potentials; for instance, the relation $\delta_j(x_s) - \sum_{x_t\in\mathcal{X}_t}\delta_j(x_s)\delta_k(x_t) = 0$ holds for all $x_s \in \mathcal{X}_s$. 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 $\theta \neq \widetilde{\theta}$). Although this many-to-one correspondence might seem undesirable, its usefulness is illustrated in Section 5.

The bulk of this paper focuses exclusively on MRFs with interactions between at most pairs (x_s, x_t) of random variables, which we refer to as *pairwise* MRFs. In principle, there is no loss of generality in restricting to pairwise interactions, since any MRF can be converted to this form by introducing auxiliary random variables; see Freeman and Weiss [22] for the details of this procedure. However, the techniques described in this paper can all be generalized to apply directly to MRFs that involve higher-order interactions, by dealing with hypertrees as opposed to ordinary trees.[1] Moreover, with the exception of specific examples involving the Ising model, we exclusively use the canonical overcomplete representation (3) defined in terms of indicator functions.

2.3 Marginal distributions on graphs

Our analysis in the sequel focuses on the local marginal distributions that are defined by the indicator functions in the canonical overcomplete representation (3). In particular, taking expectations

[1]For brevity, we do not discuss hypertrees at length in this paper. Roughly speaking, they amount to trees formed on clusters of nodes from the original graph; see Wainwright et al. [37] for further details on hypertrees.

of these indicators with respect to some distribution $p(\cdot)$ yields marginal probabilities for each node $s \in V$ and each edge $(s,t) \in E$, in the following way:

$$\mu_{s;j} \;=\; \mathbb{E}_p[\delta_j(x_s)] \;:=\; \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x})\delta_j(x_s) \qquad \text{for } s \in V,\;\; j \in \mathcal{X}_s, \tag{5a}$$

$$\mu_{st;jk} \;=\; \mathbb{E}_p[\delta_j(x_s)\delta_k(x_t)] \;:=\; \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x};\theta)\,[\delta_j(x_s)\delta_k(x_t)] \;\; \text{for } (s,t) \in E,\;\; (j,k) \in \mathcal{X}_s \times \mathcal{X}_t. \tag{5b}$$

Note that equation (5) defines a d-dimensional vector $\mu = \{\mu_\alpha, \alpha \in \mathcal{I}\}$ of marginals, indexed by elements of $\mathcal{I}$ defined in equation (4). We let $\mathrm{MARG}(G)$ denote the set of all such marginals realizable in this way:

$$\mathrm{MARG}(G) \;:=\; \{\mu \in \mathbb{R}^d \mid \mu_{s;j} = \mathbb{E}_p[\delta_j(x_s)],\;\; \mu_{st;jk} = \mathbb{E}_p[\delta_j(x_s)\delta_k(x_t)] \;\; \text{for some} \;\; p(\cdot)\;\}. \tag{6}$$

The conditions defining membership in $\mathrm{MARG}(G)$ can be expressed more compactly in the equivalent vector form $\mu = \mathbb{E}_p[\boldsymbol{\phi}(\mathbf{x})] = \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x})\boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}$ denotes a vector consisting of the potential functions forming the canonical overcomplete representation (3). We refer to $\mathrm{MARG}(G)$ as the *marginal polytope* associated with the graph G.

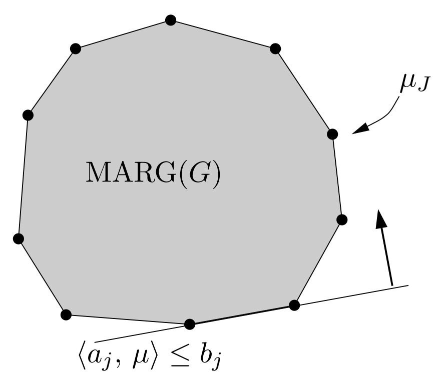


Figure 3. The geometry of a marginal polytope $\mathrm{MARG}(G)$. It is bounded and characterized by a finite number of half-space constraints $\langle a_j,\, \mu\rangle \leq b_j$. The vertices μ_J are in one-to-one correspondence with configurations $J \in \mathcal{X}^n$ in the original discrete space.

By definition, any marginal polytope is the convex hull of a finite number of vectors — namely, the collection $\{\boldsymbol{\phi}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}^n\}$. Consequently, the Minkowski-Weyl theorem [30] ensures that $\mathrm{MARG}(G)$ can be represented as an intersection of half-spaces $\cap_{j \in \mathcal{J}} H_{a_j,b_j}$, where $\mathcal{J}$ is a *finite* index set and each half-space is of the form

$$H_{a_j,b_j} \;\;:=\;\; \{\mu \in \mathbb{R}^d \mid \langle a_j,\, \mu\rangle \leq b_j\}. \tag{7}$$

for some $a_j \in \mathbb{R}^d$, and $b_j \in \mathbb{R}$. These half-space constraints include the non-negativity condition $\mu_\alpha \geq 0$ for each $\alpha \in \mathcal{I}$. Moreover, due to the overcompleteness of the canonical overcomplete representation, there are various equality[2] constraints that must hold; for instance, for all nodes $s \in V$, we have the constraint $\sum_{j \in \mathcal{X}_s} \mu_{s;j} = 1$.

Figure 3 provides an idealized illustration of the geometry of a marginal polytope, including a particular half-space constraint. A point $\mu^* \in \mathrm{MARG}(G)$ is an extreme point [30] if any decomposition of the form $\mu^* = \lambda\mu^1 + (1-\lambda)\mu^2$ and $\lambda \in (0,1)$ implies that $\mu^1 = \mu^2 = \mu^*$. It

[2]Any equality constraint $\langle a,\, \mu\rangle = b$ is equivalent to enforcing the pair of inequality constraints $\langle a,\, \mu\rangle \leq b$ and $\langle -a,\, \mu\rangle \leq -b$.

is straightforward to show that any extreme point of $\text{MARG}(G)$ is uniquely realized by a delta distribution $p_J(\mathbf{x}) := \delta_J(\mathbf{x})$ that places all its mass at a single configuration $J \in \mathcal{X}^n$. We use μ_J to denote the extreme point, or equivalently [7], the vertex associated with configuration $J \in \mathcal{X}^n$. The number of linear constraints required to characterize $\text{MARG}(G)$, though always finite, grows rapidly in n for a general graph with cycles; see Deza and Laurent [16] for an in-depth treatment of the binary case.

It is straightforward, however, to specify a *subset* of constraints that any $\mu \in \text{MARG}(G)$ must satisfy. First, as mentioned previously, since the elements of μ are marginal probabilities, we must have $\mu \geq 0$ (meaning that μ is in the positive orthant). Second, as local marginals, the elements of μ must satisfy the *normalization constraints*:

$$\sum_{j \in \mathcal{X}_s} \mu_{s;j} \;=\; 1 \quad \forall\, s \in V, \qquad\qquad \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \mu_{st;jk} \;=\; 1 \quad \forall\, (s,t) \in E. \tag{8}$$

Third, since the single node marginal over x_s must be consistent with the joint marginal on (x_s, x_t), the following *marginalization constraint* must also be satisfied:

$$\sum_{k \in \mathcal{X}_t} \mu_{st;jk} \;=\; \mu_{s;j} \quad \forall\, (s,t) \in E,\; j \in \mathcal{X}_s. \tag{9}$$

On the basis of these constraints,[3] we define the following set:

$$\text{LOCAL}(G) := \{\mu \in \mathbb{R}^d_+ \mid \sum_{j \in \mathcal{X}_s} \mu_{s;j} = 1 \;\forall\, s \in V,\; \sum_{k \in \mathcal{X}_t} \mu_{st;jk} = \mu_{s;j} \;\forall (s,t) \in E, j \in \mathcal{X}_s\}. \tag{10}$$

Here it should be understood that there are two sets of marginalization constraints for each edge (s,t): one for each of the variables x_s and x_t. By construction, $\text{LOCAL}(G)$ specifies an outer bound on $\text{MARG}(G)$; moreover, in contrast to $\text{MARG}(G)$, it involves only a number of inequalities that is polynomial in n. More specifically, $\text{LOCAL}(G)$ is defined by $\mathcal{O}(mn + m^2|E|)$ inequalities, where $m := \max_s |\mathcal{X}_s|$. Since the number of edges $|E|$ is at most $\binom{n}{2}$, this complexity is at most $\mathcal{O}(m^2 n^2)$. The constraint set $\text{LOCAL}(G)$ plays an important role in the sequel.

2.4 MAP estimation

Of central interest in this paper is the computation of *maximum a posteriori* (MAP) configurations[4] for a given distribution in an exponential form — i.e., configurations in the set $\arg\max_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}; \bar{\theta})$, where $\bar{\theta} \in \mathbb{R}^d$ is a given vector of weights. For reasons to be clarified, we refer to $p(\mathbf{x}; \bar{\theta})$ as the *target distribution*. The problem of computing an MAP configuration arises in a wide variety of applications. For example, in image processing [e.g., 8], computing MAP estimates can be used as the basis for image segmentation techniques. In error-correcting coding [e.g., 29], a decoder based on computing the MAP codeword minimizes the word error rate.

When using the canonical overcomplete representation $\boldsymbol{\phi}(\mathbf{x}) = \{\delta_j(x_s),\; \delta_j(x_s)\delta_k(x_t)\}$, it is often convenient to represent the exponential parameters in the following functional form:

$$\bar{\theta}_s(x_s) \;:=\; \sum_{j \in \mathcal{X}_s} \bar{\theta}_{s;j}\delta_j(x_s), \qquad\qquad \bar{\theta}_{st}(x_s, x_t) \;:=\; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \bar{\theta}_{st;jk}\delta_j(x_s)\delta_k(x_t). \tag{11}$$

[3] Note that the normalization constraint on $\{\mu_{st;jk}\}$ is redundant given the marginalization constraint (9), and the normalization of $\{\mu_{s;j}\}$.

[4] The term a posteriori arises from applications, in which case one often wants to compute maximizing elements of the posterior distribution $p(\mathbf{x}\,|\mathbf{y}; \theta)$, where $\mathbf{y}$ is a fixed collection of noisy observations.

With this notation, the MAP problem is equivalent to finding a configuration $\widehat{\mathbf{x}}_{\text{MAP}} \in \mathcal{X}^n$ that maximizes the quantity

$$\langle \bar{\theta}, \boldsymbol{\phi}(\mathbf{x}) \rangle \;\; := \;\; \sum_{s \in V} \bar{\theta}_s(x_s) + \sum_{(s,t) \in E} \bar{\theta}_{st}(x_s, x_t). \tag{12}$$

Although the parameter $\bar{\theta}$ is a known and fixed quantity, it is useful for analytical purposes to view it as a variable, and define a function $\Phi_\infty(\bar{\theta})$ as follows:

$$\Phi_\infty(\bar{\theta}) \;\; := \;\; \max_{\mathbf{x} \in \mathcal{X}^n} \langle \bar{\theta}, \boldsymbol{\phi}(\mathbf{x}) \rangle. \tag{13}$$

Note that $\Phi_\infty(\bar{\theta})$ represents the value of the optimal (MAP) configuration as $\bar{\theta}$ ranges over $\mathbb{R}^d$. As the maximum of a collection of linear functions, Φ_∞ is convex in terms of $\bar{\theta}$.

3 Upper bounds via convex combinations

This section introduces the basic form of the upper bounds on $\Phi_\infty(\bar{\theta})$ to be considered in this paper. The key property of Φ_∞ is its convexity, which allows us to apply Jensen's inequality [26]. More specifically, let $\{\rho^i\}$ be a finite collection of non-negative weights that sum to one, and consider a collection $\{\theta^i\}$ of exponential parameters such that $\sum_i \rho^i \theta^i = \bar{\theta}$. Then applying Jensen's inequality yields the upper bound

$$\Phi_\infty(\bar{\theta}) \;\; \leq \;\; \sum_i \rho^i \Phi_\infty(\theta^i). \tag{14}$$

Note that the bound (14) holds for any collection of exponential parameters $\{\theta^i\}$ that satisfy $\sum_i \rho^i \theta^i = \bar{\theta}$; however, the bound will not necessarily be useful, unless the evaluation of $\Phi_\infty(\theta^i)$ is easier than the original problem of computing $\Phi_\infty(\bar{\theta})$. Accordingly, in this paper, we focus on convex combinations of *tree-structured* exponential parameters (i.e., the set of non-zero components of θ^i is restricted to an acyclic subgraph of the full graph), for which exact computations are tractable. In this case, each index i in equation (14) corresponds to a spanning tree of the graph, and the corresponding exponential parameter is required to respect the structure of the tree. In the following, we introduce the necessary notation required to make this idea precise.

3.1 Convex combinations of trees

For a given graph, let T denote a particular spanning tree, and let $\mathfrak{T} = \mathfrak{T}(G)$ denote the set of all spanning trees. For a given spanning tree $T = (V, E(T))$, we define a set

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

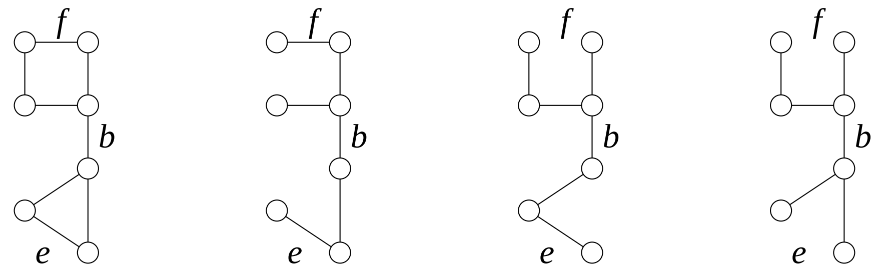
corresponding to those indexes associated with all vertices but only edges in the tree.

To each spanning tree $T \in \mathfrak{T}$, we associate an exponential parameter $\theta(T)$ that must respect the structure of T. More precisely, the parameter $\theta(T)$ must belong to the following linear constraint set:

$$\mathcal{E}(T) \;\; = \;\; \{ \; \theta(T) \in \mathbb{R}^d \mid \theta_\alpha(T) = 0 \;\; \forall \; \alpha \in \mathcal{I} \backslash \mathcal{I}(T) \; \}. \tag{15}$$

Note that $\mathcal{E}(T)$ corresponds to an e-flat manifold of exponential parameters in information-geometric terms [3]. By concatenating all of the tree vectors, we form a larger vector $\boldsymbol{\theta} = \{\theta(T), \; T \in \mathfrak{T}\}$, which is an element of $\mathbb{R}^{d \times |\mathfrak{T}(G)|}$. The vector $\boldsymbol{\theta}$ must belong to the constraint set

$$\mathcal{E} \;\; := \;\; \{\boldsymbol{\theta} \in \mathbb{R}^{d \times |\mathfrak{T}(G)|} \mid \theta(T) \in \mathcal{E}(T) \;\; \text{for all} \;\; T \; \in \; \mathfrak{T}(G)\}. \tag{16}$$



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

Figure 4. Illustration of edge appearance probabilities. 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 $\rho_f = 1/3$.

In order to define convex combinations of exponential parameters defined on spanning trees, we require a probability distribution $\boldsymbol{\rho}$ over the set of spanning trees

$$\boldsymbol{\rho} \quad := \quad \{ \, \rho(T), \; T \in \mathfrak{T} \mid \rho(T) \geq 0, \quad \sum_{T \in \mathfrak{T}} \rho(T) = 1 \, \}. \tag{17}$$

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

$$\mathrm{supp}(\boldsymbol{\rho}) \quad := \quad \{ \, T \in \mathfrak{T} \mid \rho(T) > 0 \, \}. \tag{18}$$

In the sequel, it will also be of interest to consider the probability $\rho_e = \Pr_{\boldsymbol{\rho}}\{e \in T\}$ that a given edge $e \in E$ appears in a spanning tree T chosen randomly under $\boldsymbol{\rho}$. We let $\boldsymbol{\rho_e} = \{\rho_e \mid e \in E\}$ represent a vector of these *edge appearance probabilities*. Any such vector $\boldsymbol{\rho_e}$ must belong to the so-called *spanning tree polytope* [7, 17], which we denote by $\mathbb{T}(G)$. See Figure 4 for an illustration of the edge appearance probabilities. Although we allow for the support $\mathrm{supp}(\boldsymbol{\rho})$ to be a strict subset of the set of all spanning trees, we require that $\rho_e > 0$ for all $e \in E$, so that each edge appears in at least one tree with non-zero probability.

Given a collection of tree-structured parameters $\boldsymbol{\theta}$ and a distribution $\boldsymbol{\rho}$, we form a *convex combination* of tree exponential parameters as follows

$$\mathbb{E}_{\boldsymbol{\rho}}[\theta(T)] \quad := \sum_{T} \rho(T)\theta(T). \tag{19}$$

Let $\bar{\theta} \in \mathbb{R}^d$ be the target parameter vector for which we are interested in computing Φ_∞, as well as an MAP configuration of $p(\mathbf{x}; \bar{\theta})$. For a given $\boldsymbol{\rho}$, of interest are collections $\boldsymbol{\theta}$ of tree-structured exponential parameters such that $\mathbb{E}_{\boldsymbol{\rho}}[\theta(T)] = \bar{\theta}$. Accordingly, we define the following constraint set:

$$\mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta}) \quad := \quad \{\boldsymbol{\theta} \in \mathcal{E} \mid \mathbb{E}_{\boldsymbol{\rho}}[\theta(T)] = \bar{\theta}\}. \tag{20}$$

It can be seen that $\mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$ is never empty as long as $\rho_e > 0$ for all edges $e \in E$. We say that any member $\boldsymbol{\theta}$ of $\mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$ specifies a $\boldsymbol{\rho}$*-reparameterization* of $p(\mathbf{x}; \bar{\theta})$.

Example 2 (Single cycle). To illustrate these definitions, consider a binary vector $\mathbf{x} \in \{0,1\}^4$ on a 4-node cycle, with the distribution in the minimal Ising form

$$p(\mathbf{x}; \bar{\theta}) \quad \propto \quad \exp\{x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1\}.$$

In words, the target distribution is specified by the minimal parameter $\bar{\theta} \; = [0\ 0\ 0\ 0\ \ 1\ 1\ 1\ 1]$, where the zeros represent the fact that $\bar{\theta}_s = 0$ for all $s \in V$. The four possible spanning trees

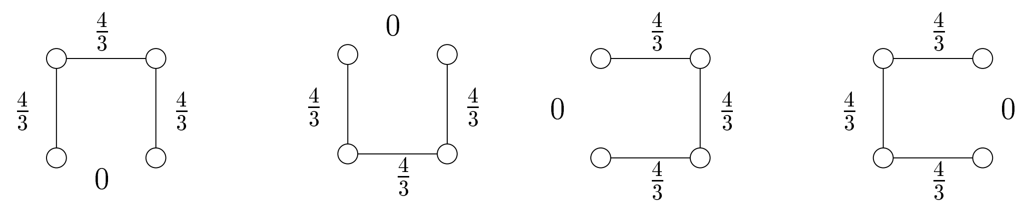


Figure 5. A convex combination of four distributions $p(\mathbf{x}; \theta(T_i))$, each defined by a spanning tree T_i, is used to approximate the target distribution $p(\mathbf{x}; \bar{\theta})$ on the single-cycle graph.

$\mathfrak{T} = \{T_i \mid i = 1, \ldots, 4\}$ on a single 4-node cycle are illustrated in Figure 5. Suppose that $\boldsymbol{\rho}$ is the uniform distribution $\rho(T_i) = 1/4$ for $i = 1, \ldots 4$, so that $\rho_e = 3/4$ for each edge $e \in E$. We construct a member $\boldsymbol{\theta}$ of $\mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$, as follows:

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

With this choice, it is easily verified that $\mathbb{E}_{\boldsymbol{\rho}}[\theta(T)] = \bar{\theta}$ so that $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$.

3.2 Tightness of upper bounds

It follows from equations (14), (19) and (20) that for any $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$, there holds:

$$\Phi_\infty(\bar{\theta}) \quad \leq \quad \sum_T \rho(T) \Phi_\infty(\theta(T)) \;\; = \;\; \sum_T \rho(T) \max_{\mathbf{x} \in \mathcal{X}^n} \big\{ \langle \theta(T),\, \boldsymbol{\phi}(\mathbf{x}) \rangle \big\}. \tag{21}$$

Our first goal is to understand when the upper bound (21) is *tight* — that is, met with equality. It turns out that that equality holds if and only if the collection of trees share a common optimum, which leads to the notion of *tree agreement*.

More formally, for any exponential parameter vector $\theta \in \mathbb{R}^d$, define the collection of its optimal configurations:

$$\mathrm{OPT}(\theta) \quad := \quad \{\mathbf{x} \in \mathcal{X}^n \mid \langle \theta,\, \boldsymbol{\phi}(\mathbf{x}') \rangle \; \leq \langle \theta,\, \boldsymbol{\phi}(\mathbf{x}) \rangle \;\; \text{for all } \mathbf{x}' \in \mathcal{X}^n \}. \tag{22}$$

Note that by the definition (13) of Φ_∞, there holds $\langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle = \Phi_\infty(\bar{\theta})$ for any $\mathbf{x} \in \mathrm{OPT}(\bar{\theta})$. With this notation, we have:

Proposition 1 (Tree agreement). *Let $\boldsymbol{\theta} = \{\theta(T)\} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$, and let $\cap_{T \in \mathrm{supp}(\boldsymbol{\rho})} \mathrm{OPT}(\theta(T))$ be the set of configurations that are optimal for every tree-structured distribution. Then the following containment always holds:*

$$\cap_{T \in \mathrm{supp}(\boldsymbol{\rho})} \mathrm{OPT}(\theta(T)) \quad \subseteq \quad \mathrm{OPT}(\bar{\theta}). \tag{23}$$

Moreover, the bound (21) is tight if and only if the intersection on the LHS is non-empty.

Proof. The containment relation is clear from the form of the upper bound (21). Let $\mathbf{x}^*$ belong to $\mathrm{OPT}(\bar{\theta})$. Then the difference of the RHS and the LHS of equation (21) can be written as follows:

$$0 \;\leq\; \Big[\sum_T \rho(T)\Phi_\infty(\theta(T))\Big] - \Phi_\infty(\bar{\theta}) \;=\; \Big[\sum_T \rho(T)\Phi_\infty(\theta(T))\Big] - \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}^*)\rangle$$
$$= \sum_T \rho(T)\big[\Phi_\infty(\theta(T)) - \langle \theta(T),\, \boldsymbol{\phi}(\mathbf{x}^*)\rangle\big],$$

where the last equality uses the fact that $\sum_T \rho(T)\theta(T) = \bar{\theta}$. Now for each $T \in \mathrm{supp}(\boldsymbol{\rho})$, the term $\Phi_\infty(\theta(T)) - \langle \theta(T),\, \boldsymbol{\phi}(\mathbf{x}^*)\rangle$ is non-negative, and equal to zero only when $\mathbf{x}^*$ belongs to $\mathrm{OPT}(\theta(T))$. Therefore, the bound is met with equality if and only if $\mathbf{x}^* \in \cap_{T \in \mathrm{supp}(\boldsymbol{\rho})} \mathrm{OPT}(\theta(T))$ for some $\mathbf{x}^* \in \mathrm{OPT}(\bar{\theta})$. $\square$

The preceding result shows that the upper bound (21) is tight if and only if all the trees in the support of $\boldsymbol{\rho}$ agree on a common configuration. When this *tree agreement* condition holds, a MAP configuration for the original problem $p(\mathbf{x};\bar{\theta})$ can be obtained simply by examining the intersection $\cap_{T \in \mathrm{supp}(\boldsymbol{\rho})} \mathrm{OPT}(\theta(T))$ of configurations that are optimal on every tree for which $\rho(T) > 0$. Accordingly, we focus our attention on the problem of finding upper bounds (21) that are tight, so that a MAP configuration can be obtained. Since the target parameter $\bar{\theta}$ is fixed and assuming that we fix the spanning tree distribution $\boldsymbol{\rho}$, the problem on which we focus is that of optimizing the upper bound as a function of $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$. Proposition 1 suggests two different strategies for attempting to find a tight upper bound, which are the subjects of the next two sections:

Direct minimization and Lagrangian duality: The first approach is a direct one, based on minimizing equation (21). In particular, for a fixed distribution $\boldsymbol{\rho}$ over spanning trees, we consider the constrained optimization problem of minimizing the RHS of equation (21) subject to the constraint $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$. The problem structure ensures that strong duality holds, so that it can be tackled via its Lagrangian dual. In Section 4, we show that this dual problem is a linear programming (LP) relaxation of the original MAP estimation problem.

Message-passing approach: In Section 5, we derive and analyze message-passing algorithms, the goal of which is to find, for a fixed distribution $\boldsymbol{\rho}$, a collection of exponential parameters $\boldsymbol{\theta}^* = \{\theta^*(T)\}$ such that $\boldsymbol{\theta}^*$ belongs to the constraint set $\mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$ of equation (20), and the intersection $\cap_T \mathrm{OPT}(\theta^*(T))$ of configurations optimal for all tree problems is non-empty. Under these conditions, Proposition 1 guarantees that for all configurations in the intersection, the bound is tight. In Section 5, we develop a class of message-passing algorithms with these two goals in mind. We also prove that when the bound is tight, fixed points of these algorithms specify optimal solutions to the LP relaxation derived in Section 4.

4 Lagrangian duality and tree relaxation

In this section, we develop and analyze a Lagrangian reformulation of the problem of optimizing the upper bounds —- i.e., minimizing the RHS of equation (21) as a function of $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$. The cost function is a linear combination of convex functions, and so is also convex as a function of $\boldsymbol{\theta}$; moreover, the constraints are linear in $\boldsymbol{\theta}$. Therefore, the minimization problem can be solved via its Lagrangian dual. Before deriving this dual, it is convenient to develop an alternative representation of Φ_∞ as a linear program.

4.1 Linear program over the marginal polytope for exact MAP estimation

Recall from equation (13) that the function value $\Phi_\infty(\bar{\theta})$ corresponds to the optimal value of the integer program $\max_{\mathbf{x}} \langle \bar{\theta}, \boldsymbol{\phi}(\mathbf{x}) \rangle$. We now reformulate this integer program as a linear program (LP), which leads to an alternative representation of the function Φ_∞, and hence of the exact MAP estimation problem. In order to convert from integer to linear program, our approach is the standard one [e.g., 7] of taking convex combinations of all possible solutions. The resulting convex hull is precisely the marginal polytope $\mathrm{MARG}(G)$ defined in Section 2.3. We summarize in the following:

Lemma 1. *The function $\Phi_\infty(\bar{\theta})$ has an alternative representation as a linear program over the marginal polytope:*

$$\Phi_\infty(\bar{\theta}) \;=\; \max_{\mu \in \mathrm{MARG}(G)} \langle \bar{\theta},\, \mu \rangle \;=\; \max_{\mu \in \mathrm{MARG}(G)} \Big\{ \sum_{s \in V} \sum_{j} \mu_{s;j} \bar{\theta}_{s;j} + \sum_{(s,t) \in E} \; \sum_{(j,k)} \mu_{st;jk} \bar{\theta}_{st;jk} \Big\}. \tag{24}$$

Proof. Consider the set $\mathcal{P} := \{\, p(\cdot) \mid p(\mathbf{x}) \geq 0, \;\; \sum_{\mathbf{x}} p(\mathbf{x}) = 1\}$ of all possible probability distributions over $\mathbf{x}$. We first claim that the maximization over $\mathbf{x} \in \mathcal{X}^n$ can be rewritten as an equivalent maximization over $\mathcal{P}$ as follows:

$$\max_{\mathbf{x} \in \mathcal{X}^n} \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle \;=\; \max_{p \in \mathcal{P}} \Big\{ \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}) \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle \Big\}. \tag{25}$$

On one hand, the RHS is certainly greater than or equal to the LHS, because for any configuration $\mathbf{x}^*$, the set $\mathcal{P}$ includes the delta distribution that places all its mass at $\mathbf{x}^*$. On the other hand, for any $p \in \mathcal{P}$, the sum $\sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}) \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle$ is a convex combination of terms of the form $\langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle$ for $\mathbf{x} \in \mathcal{X}^n$, and so cannot be any larger than $\max_{\mathbf{x}} \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle$.

Making use of the functional notation in equation (11), we now expand the summation on the RHS of equation (25), and then use the linearity of expectation to write:

$$\sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}) \Big\{ \sum_{s \in V} \bar{\theta}_s(x_s) + \sum_{(s,t) \in E} \bar{\theta}_{st}(x_s, x_t) \Big\} \;=\; \sum_{s \in V} \sum_{j \in \mathcal{X}_s} \bar{\theta}_{s;j} \mu_{s;j} + \sum_{(s,t) \in E} \; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \bar{\theta}_{st;jk} \mu_{st;jk}.$$

Here $\mu_{s;j} := \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}) \delta_j(x_s)$ and $\mu_{st;jk} := \sum_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x}) \delta_{jk}(x_s, x_t)$. As p ranges over $\mathcal{P}$, the marginals μ range over $\mathrm{MARG}(G)$. Therefore, we conclude that $\max_{\mathbf{x} \in \mathcal{X}^n} \langle \bar{\theta},\, \boldsymbol{\phi}(\mathbf{x}) \rangle$ is equal to $\max_{\mu \in \mathrm{MARG}(G)} \langle \bar{\theta},\, \mu \rangle$, as claimed. $\square$

Remarks: (a) Lemma 1 identifies $\Phi_\infty(\bar{\theta})$ as the *support function* [26] of the set $\mathrm{MARG}(G)$. In particular, consider the family of hyperplanes specified by the normal vector $\bar{\theta} \in \mathbb{R}^d$, and varying intercepts. By the definition of Φ_∞, the inequality $\langle \bar{\theta},\, \mu \rangle - \Phi_\infty(\bar{\theta}) \leq 0$ holds for all $\mu \in \mathrm{MARG}(G)$, with equality for at least some member of $\mathrm{MARG}(G)$. Therefore, $\Phi_\infty(\bar{\theta})$ can be interpreted as the negative intercept of the supporting hyperplane to $\mathrm{MARG}(G)$ with normal vector $\bar{\theta} \in \mathbb{R}^d$. This property underlies the dual relation that is the focus of Section 4.2.

(b) The equivalence asserted in Lemma 1 can be understood geometrically by referring back to Figure 3, which provides an idealized illustration of a marginal polytope $\mathrm{MARG}(G)$. Recall that the vertices of $\mathrm{MARG}(G)$, denoted by μ_J, are in one-to-one correspondence with configurations $J \in \mathcal{X}^n$. An important property of linear programs is that the optimum is always attained at a vertex (possibly more than one) of the associated constraint set [e.g., 7]. Consequently, solving the LP $\max_{\mu \in \mathrm{MARG}(G)} \langle \bar{\theta},\, \mu \rangle$ will always yield a vertex μ_J of $\mathrm{MARG}(G)$, which indicates that the configuration $J \in \mathcal{X}^n$ is an MAP configuration of $p(\mathbf{x};\bar{\theta})$. The difficulty with this LP formulation of exact MAP estimation is that MARG(G) is a very complicated polytope, with a large number of facets [e.g., 16].

4.2 Lagrangian dual

Let us now address the problem of finding the tightest upper bound of the form in equation (21). More formally, for a fixed distribution $\boldsymbol{\rho}$ over spanning trees, we wish to solve the constrained optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{\theta}\in\mathcal{E}} & \sum_T \rho(T)\Phi_\infty(\theta(T)) \\ \text{s. t} & \sum_T \rho(T)\theta(T) = \bar{\theta}. \end{cases} \tag{26}$$

As defined in equation (16), the constraint set $\mathcal{E}$ consists of all vectors $\boldsymbol{\theta} = \{\theta(T)\}$ such that for each tree T, the subvector $\theta(T)$ respects the structure of T, meaning that $\theta_\alpha(T) = 0 \;\; \forall\, \alpha \in \mathcal{I}\backslash\mathcal{I}(T)$.

Note that the cost function is a convex combination of convex functions; moreover, with $\boldsymbol{\rho}$ fixed, the constraints are all linear in $\boldsymbol{\theta}$. Under these conditions, strong duality holds [6], so that this constrained optimization problem can be tackled via its Lagrangian dual. The dual LP formulation turns out to have a surprisingly simple and intuitive form. In particular, recall the set LOCAL(G) described in Section 2.3. For the convenience of the reader, we restate the definition here:

$$\text{LOCAL}(G) := \{\tau \in \mathbb{R}^d_+ \mid \sum_{j\in\mathcal{X}_s} \tau_{s;j} = 1 \;\forall\; s \in V, \;\; \sum_{k\in\mathcal{X}_t} \tau_{st;jk} = \tau_{s;j} \;\forall (s,t) \in E, j \in \mathcal{X}_s\}. \tag{27}$$

By construction, this set is an outer bound on MARG(G), since any valid marginal vector must satisfy all of the constraints defining LOCAL(G). However, for a graph with cycles, LOCAL(G) is a strict superset of MARG(G). (In particular, Example 3 to follow provides an explicit construction of an element $\tau \in$ LOCAL$(G)\backslash$ MARG(G).) For this reason, we call any $\tau \in$ LOCAL(G) a *pseudomarginal* vector.

The structure of the polytope LOCAL(G), in contrast to that of MARG(G), is very simple for any graph: indeed, it is characterized by $\mathcal{O}(mn + m^2|E|)$ constraints, where $m := \max_s |\mathcal{X}_s|$. A natural relaxation of the linear program in equation (24), then, is to replace the constraint set MARG(G) by the outer bound LOCAL(G). It turns out that the Lagrangian dual of problem (26) is precisely this relaxed linear program:

Theorem 1 (Lagrangian dual). *The Lagrangian dual to problem* (26) *is given by the LP relaxation:*

$$\Phi_\infty(\bar{\theta}) \;\; \leq \;\; \max_{\tau\in\text{LOCAL}(G)} \langle \tau,\, \bar{\theta}\rangle \;\; \equiv \;\; \max_{\tau\in\text{LOCAL}(G)} \big\{ \sum_{s\in V}\sum_{j} \tau_{s;j}\bar{\theta}_{s;j} + \sum_{(s,t)\in E} \; \sum_{(j,k)} \tau_{st;jk}\bar{\theta}_{st;jk} \big\}. \tag{28}$$

Proof. Let τ be a vector of Lagrange multipliers corresponding to the constraints $\mathbb{E}_{\boldsymbol{\rho}}[\theta(T)] = \bar{\theta}$. We then form the Lagrangian associated with problem (26):

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\rho},\bar{\theta}}(\boldsymbol{\theta}, \tau) &= \mathbb{E}_{\boldsymbol{\rho}}[\Phi_\infty(\theta(T))] + \langle \tau,\, \bar{\theta} - \sum_T \rho(T)\theta(T)\rangle \\ &= \sum_T \rho(T)\, \big[\Phi_\infty(\theta(T)) - \langle \theta(T),\, \tau\rangle\big] \; + \; \langle \tau,\, \bar{\theta}\rangle. \end{aligned}$$

We compute the dual by taking the infimum over $\boldsymbol{\theta} \in \mathcal{E}$, for a fixed dual vector τ:

$$\mathcal{Q}_{\boldsymbol{\rho},\bar{\theta}}(\tau) \;\; := \;\; \inf_{\boldsymbol{\theta}\in\mathcal{E}} \mathcal{L}_{\boldsymbol{\rho},\bar{\theta}}(\boldsymbol{\theta}, \tau) \; = \; \sum_T \rho(T) \inf_{\theta(T)\in\mathcal{E}(T)} \big[\Phi_\infty(\theta(T)) - \langle \theta(T),\, \tau\rangle\big] \; + \; \langle \tau,\, \bar{\theta}\rangle. \tag{29}$$

The following lemma is key to computing this infimum:

Lemma 2 (Indicator function). *We have the following conjugate dual relation:*

$$\sup_{\theta(T) \in \mathcal{E}(T)} \left\{ \langle \theta(T),\, \tau \rangle - \Phi_\infty(\theta(T)) \right\} \;=\; \begin{cases} 0 & \text{if} \quad \tau \in \text{LOCAL}(G;T) \\ +\infty & \text{otherwise,} \end{cases}$$

where $\text{LOCAL}(G;T) := \{\, \tau \in \mathbb{R}^d_+ \mid \sum_{j \in \mathcal{X}_s} \tau_{s;j} = 1 \;\forall s \in V,\;\; \sum_{j \in \mathcal{X}_s} \tau_{st;jk} = \tau_{t;k} \;\forall\, (s,t) \in E(T) \,\}$.

Proof. See Appendix A. □

Using Lemma 2, the value of the infimum (29) will be equal to $\langle \tau, \bar{\theta} \rangle$ if $\tau \in \text{LOCAL}(G;T)$ for all $T \in \text{supp}(\boldsymbol{\rho})$, and $-\infty$ otherwise. Since every edge in the graph belongs to at least one tree in $\text{supp}(\boldsymbol{\rho})$, we have $\cap_{T \in \text{supp}(\boldsymbol{\rho})} \text{LOCAL}(G;T) \;\equiv\; \text{LOCAL}(G)$, so that the dual function takes the form:

$$\mathcal{Q}_{\boldsymbol{\rho},\bar{\theta}}(\tau) \;=\; \begin{cases} \langle \tau, \bar{\theta} \rangle & \text{if } \tau \in \text{LOCAL}(G) \\ -\infty & \text{otherwise.} \end{cases}$$

Thus, the dual optimal value is $\max_{\tau \in \text{LOCAL}(G)} \langle \tau, \bar{\theta} \rangle$; by strong duality [6], this optimum is equal to the optimal primal value (26). □

Equation (28) is a relaxation of the original LP (24), since it involves replacing the original constraint set $\text{MARG}(G)$ by the set $\text{LOCAL}(G)$ formed by local (node and edgewise) constraints. In fact, the polytope $\text{LOCAL}(G)$ is equivalent to the constraint set involved in the Bethe variational principle which, as shown by Yedidia et al. [39], underlies the sum-product algorithm. For binary variables, the linear program (28) is equivalent to a relaxation that has been studied in previous work [e.g., 11, 16, 25, 31]. The derivation given here illuminates the critical role of graphical structure in controlling the tightness of such a relaxation. In particular, an immediate consequence of our development is the following:

Corollary 1. *The relaxation* (28) *is exact for any problem on a tree-structured graph.*

Since the LP relaxation (28) is always exact for MAP estimation with any tree-structured distribution, we refer to it as a *tree relaxation*. For a graph with cycles — in sharp contrast to the tree-structured case —- $\text{LOCAL}(G)$ is a strict outer bound on $\text{MARG}(G)$, and the relaxation (28) will not always be tight. Figure 6 provides an idealized illustration of $\text{LOCAL}(G)$, and its relation to the exact marginal polytope $\text{MARG}(G)$. Recall from Section 2.3 that the vertices of $\text{MARG}(G)$ are all of the form μ_J, corresponding to the marginal vector realized by the delta distribution that puts all its mass on $J \in \mathcal{X}^n$. In the canonical overcomplete representation (3), each element of any such μ_J is either zero or one. These *integral* vertices, denoted by μ_{int}, are drawn with black circles in Figure 6(a). It is straightforward to show that each such μ_J is also a vertex of $\text{LOCAL}(G)$. However, for graphs with cycles, $\text{LOCAL}(G)$ includes additional *fractional* vertices that lie strictly outside of $\text{MARG}(G)$, and that are drawn in gray circles in Figure 6(a).

Since $\text{LOCAL}(G)$ is also a polytope, the optimum of the LP relaxation (28) will be attained at a vertex (possibly more than one) of $\text{LOCAL}(G)$. In geometrical terms, the cost vector $\bar{\theta}$ specifies a direction. The linear program can be solved by translating the hyperplane with normal $\bar{\theta}$ until it is tangent to the constraint set. Any vertex involved in the tangency, which need not be unique in general, corresponds to an optimizing point. This geometric construction is illustrated for two possible cost vectors in panels (b) and (c) of Figure 6.

Therefore, solving the LP relaxation using $\text{LOCAL}(G)$ as an outer bound on $\text{MARG}(G)$ can have one of two possible outcomes:

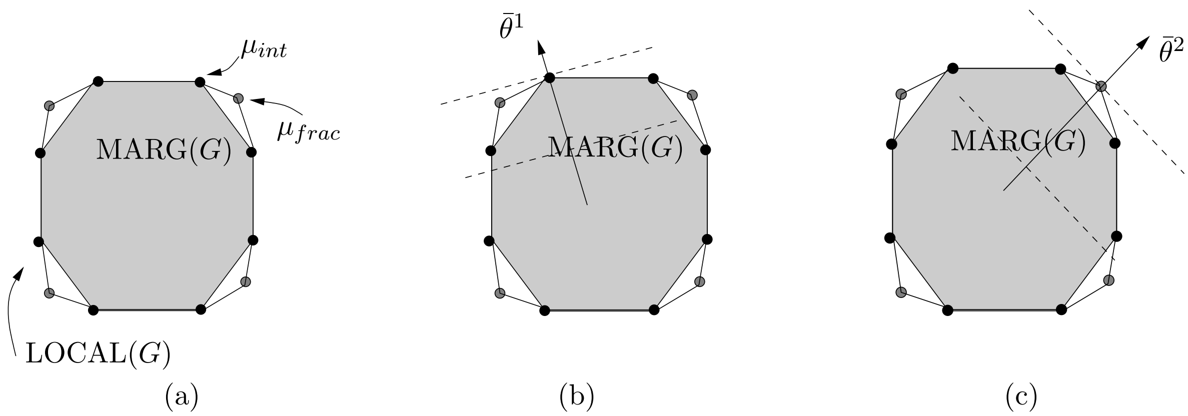


Figure 6. (a) The constraint set LOCAL(G) is an outer bound on the exact marginal polytope. Its vertex set includes all the integral vertices of MARG(G), which are in one-to-one correspondence with optimal solutions of the integer program. It also includes additional fractional vertices, which are *not* vertices of MARG(G). (b)– (c) Solving a LP with cost vector $\bar{\theta}$ entails translating a hyperplane with normal $\bar{\theta}$ until it is tangent to the constraint set LOCAL(G). In (b), the point of tangency occurs at a unique integral vertex. In (c), the tangency occurs at a fractional vertex of LOCAL(G) that lies outside of MARG(G).

(i) First, suppose that the optimum is attained at a vertex of LOCAL(G) which is also a vertex of MARG(G). The optimum may occur at a unique integral vertex, as illustrated in panel (b), or at multiple integral vertices (not illustrated here). In this case, both the dual LP relaxation (28), and hence also the primal version in equation (21), are tight. As observed previously, vertices of MARG(G) are in one-to-one correspondence with configurations $J \in \mathcal{X}^n$. Therefore, in this case, we can recover an optimal MAP configuration for the original problem, which is consistent with Proposition 1.

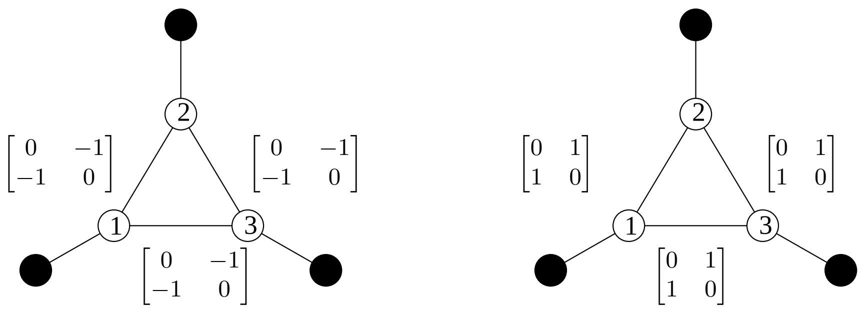
(ii) Alternatively, the optimum is attained only outside the original marginal polytope MARG(G) at a fractional vertex of LOCAL(G), as illustrated in panel (c). In this case, the relaxation must be loose, so that Proposition 1 asserts that it is impossible to find a configuration that is optimal for all tree-structured problems.

Consequently, whether or not the tree agreement condition of Proposition 1 can be satisfied corresponds precisely to the distinction between integral and fractional vertices in the LP relaxation (28).

Example 3 (Integral versus fractional vertices). In order to demonstrate explicitly the distinction between fractional and integral vertices, we now consider a simple example — namely, a binary problem $\mathbf{x} \in \{0,1\}^3$ defined the 3-node cycle illustrated in Figure 7. Consider the parameter vector $\bar{\theta}$ with components defined as follows:

$$\bar{\theta}_s \;=\; \begin{bmatrix}\bar{\theta}_{s;0} & \bar{\theta}_{s;1}\end{bmatrix} \;:=\; \begin{bmatrix}0 & 0\end{bmatrix} \;\forall\, s, \qquad \bar{\theta}_{st} \;=\; \begin{bmatrix}\bar{\theta}_{st;00} & \bar{\theta}_{st;01} \\ \bar{\theta}_{st;10} & \bar{\theta}_{st;11}\end{bmatrix} \;:=\; \begin{bmatrix}0 & -\beta \\ -\beta & 0\end{bmatrix} \quad \forall\,(s,t). \tag{30}$$

Suppose first that β is positive — say $\beta = 1$ as illustrated in Figure 7(a). By construction of $\bar{\theta}$, we have $\langle \bar{\theta},\, \tau\rangle \leq 0$ for all $\tau \in$ LOCAL(G). This inequality is tight when τ is either the vertex μ_0 corresponding to the configuration [0 0 0], or its counterpart μ_1 corresponding to [1 1 1]. In fact, both of these configurations are MAP-optimal for the original problem, so that we conclude that the LP relaxation (28) is tight (i.e., we can achieve tree agreement).



(a) (b)

Figure 7. Failure of tree agreement. (a) Case $\beta > 0$ for which both [0 0 0] and [1 1 1] are MAP-optimal configurations. The relaxation (28) is tight in this case. (b) Case $\beta < 0$ for which the relaxation fails.

On the other hand, suppose that $\beta < 0$; for concreteness, say $\beta = -1$, as illustrated in Figure 7(b). This choice of $\bar{\theta}$ encourages all pairs of configurations (x_s, x_t) to be distinct (i.e., $x_s \neq x_t$). However, in going around the cycle, there must hold $x_s = x_t$ for at least one pair. Therefore, the set of optimal configurations consists of [1 0 1], and the other five permutations thereof. (I.e., all configurations except [1 1 1] and [0 0 0] are optimal). The value of any such optimizing configuration — i.e., $\max_{\mu \in \text{MARG}(G)} \langle \bar{\theta}, \mu \rangle$ — will be $-2\beta > 0$, corresponding to the fact that two of the three pairs are distinct.

However, with reference to the relaxed polytope LOCAL(G), a larger value of $\langle \bar{\theta}, \tau \rangle$ can be attained. We begin by observing that $\langle \bar{\theta}, \tau \rangle \leq -3\beta$ for all $\tau \in \text{LOCAL}(G)$. In fact, equality can be achieved in this inequality by the following pseudomarginal:

$$\tau_s = \begin{bmatrix} \tau_{s;0} & \tau_{s;1} \end{bmatrix} := \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \ \forall\, s, \qquad \tau_{st} = \begin{bmatrix} \tau_{st;00} & \tau_{st;01} \\ \tau_{st;10} & \tau_{st;11} \end{bmatrix} := \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \quad \forall\, (s,t). \quad (31)$$

Overall, we have shown that $\max_{\tau \in \text{LOCAL}(G)} \langle \bar{\theta}, \tau \rangle = -3\beta > -2\beta = \max_{\mu \in \text{MARG}(G)} \langle \bar{\theta}, \mu \rangle$, which establishes that the relaxation (28) is loose for this particular problem. Moreover, the pseudomarginal vector τ defined in equation (31) corresponds to a fractional vertex of LOCAL(G), so that we are in the geometric setting of Figure 6(c). $\diamondsuit$

5 Tree-reweighted message-passing algorithms

The main result of the preceding section is that the problem of finding tight upper bounds, as formulated in equation (26), is equivalent to solving the relaxed linear program (28) over the constraint set LOCAL(G). A key property of this constraint set is that it is defined by a number of constraints that is at most quadratic in the number of nodes n. Solving an LP over LOCAL(G), then, is certainly feasible by various generic methods, including the simplex algorithm [e.g., 7]. It is also of interest to develop algorithms that exploit the graphical structure intrinsic to the problem. Accordingly, this section is devoted to the development of iterative algorithms with this property. An interesting property of the iterative methods developed here is that when applied to a tree-structured graph, they all reduce to the ordinary max-product algorithm [22, 36]. For graphs with cycles, in contrast, they differ from the ordinary max-product algorithm in critical ways. Ultimately, we connect fixed points of these iterative algorithms to optimal dual solutions of the LP relaxation (28).

We begin with some background on the notion of *max-marginals*, and their utility in computing exact MAP estimates of tree-structured distributions [2, 14, 15, 36]. We then define an analogous notion of *pseudo-max-marginals* for graphs with cycles, which play a central role in the message-passing algorithms that we develop subsequently.

5.1 Max-marginals for tree-distributions

Although the notion of max-marginal can be defined for any distribution, of particular interest in the current context are the max-marginals associated with a distribution $p(\mathbf{x};\theta(T))$ that is Markov with respect to some tree $T = (V, E(T))$. For each $s \in V$ and $j \in \mathcal{X}_s$, the associated single node max-marginal is defined by a maximization over all other nodes in the graph

$$\nu_{s;j} \quad := \quad \kappa_s \max_{\{\mathbf{x} \mid x_s = j\}} p(\mathbf{x};\theta(T)). \tag{32}$$

Here $\kappa_s > 0$ is some normalization constant, included for convenience, that is independent of j but can vary from node to node. Consequently, the max-marginal $\nu_{s;j}$ is proportional to the probability of the most likely configuration under the constraint $x_s = j$. Note that $\nu_{s;j}$ is obtained by maximizing over the random variables at all nodes $t \neq s$, whence the terminology "max-marginal". For each edge (s,t), the joint pairwise max-marginal is defined in an analogous manner:

$$\nu_{st;jk} \quad := \quad \kappa_{st} \max_{\{\mathbf{x} \mid (x_s,x_t)=(j,k)\}} p(\mathbf{x};\theta(T)). \tag{33}$$

Once again, the quantity κ_{st} is a positive normalization constant that can vary from edge to edge but does not depend on (j,k).

It is convenient to represent all the values $\{\nu_{s;j}, j \in \mathcal{X}_s\}$ associated with a given node, and the values $\{\nu_{st;jk}, (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}$ associated with a given edge in the functional form:

$$\nu_s(x_s) \;:=\; \sum_{j \in \mathcal{X}_s} \nu_{s;j}\delta_j(x_s), \qquad \nu_{st}(x_s,x_t) \;:=\; \sum_{(j,k)\in\mathcal{X}_s\times\mathcal{X}_t} \nu_{st;jk}\delta_j(x_s)\delta_k(x_t). \tag{34}$$

It is well-known [14] that any tree-structured distribution $p(\mathbf{x};\theta(T))$ can be factorized in terms of its max-marginals as follows:

$$p(\mathbf{x};\theta(T)) \quad \propto \quad \prod_{s\in V}\nu_s(x_s) \prod_{(s,t)\in E(T)} \frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)\nu_t(x_t)}. \tag{35}$$

This factorization, which is entirely analogous to the more familiar one in terms of (sum)-marginals, is a special case of the more general junction tree decomposition [15, 14]. Moreover, it can be shown [15, 36] that the ordinary max-product (min-sum) algorithm computes this max-marginal factorization. The fact that this factorization can be computed in a straightforward manner for any tree is exploited in the algorithms that we develop in the sequel.

The max-marginal factorization (35) yields a local criterion for assessing the validity of tree max-marginals. The following lemma provides a precise statement:

Lemma 3. *A collection $\{\nu_s, \nu_{st}\}$ are valid max-marginals for a tree if and only if the edgewise consistency condition*

$$\nu_s(x_s) \quad = \quad \kappa \max_{x'_t \in \mathcal{X}_t} \nu_{st}(x_s, x'_t) \tag{36}$$

holds[5] *for every edge $(s,t) \in E(T)$.*

[5] Here κ is a positive constant that depends on both the edge, and the variable over which the maximization takes place.

Proof. Necessity of the edge consistency is clear. The sufficiency can be established by an inductive argument in which successive nodes are stripped from the tree by local maximization; see [15, 36] for further details. □

The max-marginal representation (35) allows the global problem of MAP estimation to be solved by performing a set of local maximization operations. In particular, suppose that the configuration $\mathbf{x}^*$ belongs to $\mathrm{OPT}(\theta(T))$, meaning that is MAP-optimal for $p(\mathbf{x}; \theta(T))$. For a tree, such configurations are completely characterized by *local optimality conditions* with respect to the max-marginals, as summarized in the following:

Lemma 4 (Local optimality). *Let $\{\nu_s, \nu_{st}\}$ be a valid set of max-marginals for a tree-structured graph. Then a configuration $\mathbf{x}^*$ belongs to $\mathrm{OPT}(\theta(T))$ if and only if the following local optimality conditions hold:*

$$x_s^* \in \arg\max_{x_s} \nu_s(x_s) \quad \forall \quad s, \tag{37a}$$

$$(x_s^*, x_t^*) \in \arg\max_{x_s, x_t} \nu_{st}(x_s, x_t) \quad \forall \quad (s,t) \tag{37b}$$

Proof. The necessity of the conditions in equation (37) is clear. To establish sufficiency, we follow a dynamic-programming procedure. Any tree can be rooted at a particular node $r \in V$, and all edges can be directed from parent to child $(s \to t)$. To find a configuration $\mathbf{x}^* \in \mathrm{OPT}(\theta(T))$, begin by choosing an element $x_r^* \in \arg\max_{x_r} \nu_r(x_r)$. Then proceed recursively down the tree, from parent s to child t, at each step choosing the child configuration x_t^* from $\arg\max_{x_t} \nu_{st}(x_s^*, x_t)$. By construction, the configuration $\mathbf{x}^*$ so defined is MAP-optimal; see [15, 36] for further details. □

A particularly simple condition under which the local optimality conditions (37) hold is when for each $s \in V$, the max-marginal ν_s has a *unique optimum* x_s^*. In this case, the MAP configuration $\mathbf{x}^*$ is unique with elements $x_s^* = \arg\max_{x_s} \nu_s(x_s)$ that are computed easily by maximizing each single node max-marginal. If this uniqueness condition does not hold, then more than one configuration is MAP-optimal for $p(\mathbf{x}; \bar{\theta})$. In this case, maximizing each single node max-marginal is no longer sufficient [36], and the dynamic-programming procedure described in the proof of Lemma 4 must be used.

5.2 Iterative algorithms

We now turn to the development of iterative algorithms for a graph $G = (V, E)$ that contains cycles. We begin with a high-level overview of the concepts and objectives, before proceeding to a precise description.

5.2.1 High level view

The notion of max-marginal is not limited to distributions defined by tree-structured graphs, but can also be defined for graphs with cycles. Indeed, if we were able to compute the exact max-marginals of $p(\mathbf{x}; \bar{\theta})$ and each single node max-marginal had a unique optimum, then the MAP estimation problem could be solved by local optimizations.[6] However, computing exact max-marginals for a distribution on a general graph with cycles is an intractable task. Therefore, it is again necessary to relax our requirements.

The basic idea, then, is to consider a vector of so-called *pseudo-max-marginals* $\nu := \{\nu_s, \nu_{st}\}$, the properties of which are to be defined shortly. The qualifier "pseudo" reflects the fact that these

[6]If a subset of the single node max-marginals had multiple optima, the situation would be more complicated.

quantities no longer have an interpretation as exact max-marginals, but instead represent approximations to max-marginals on the graph with cycles. For a given distribution $\boldsymbol{\rho}$ over the spanning trees of the graph G and a tree T for which $\rho(T) > 0$, consider the subset $\nu(T)$, corresponding to those elements of ν associated with T — i.e.,

$$\nu(T) \;\; := \;\; \{\nu_s, \; s \in V\} \; \cup \; \{\nu_{st}, \; (s,t) \in E(T)\}. \tag{38}$$

We think of $\nu(T)$ as implicitly specifying a tree-structured exponential parameter via the factorization (35), i.e.:

$$\nu(T) \;\; \longleftrightarrow \;\; \theta(T), \quad \forall \, T \in \operatorname{supp}(\boldsymbol{\rho}), \tag{39}$$

which in turn implies that ν is associated with a collection of tree-structured parameters — viz.:

$$\nu \;\; \longleftrightarrow \;\; \boldsymbol{\theta} := \{\theta(T) \mid T \in \operatorname{supp}(\boldsymbol{\rho})\}. \tag{40}$$

Now suppose that given $\boldsymbol{\rho}$, we have a vector ν that satisfies the following properties:

(a) The vector ν specifies a vector $\boldsymbol{\theta} \in \mathcal{A}_{\boldsymbol{\rho}}(\bar{\theta})$, meaning that $\boldsymbol{\theta}$ is a $\boldsymbol{\rho}$*-reparameterization* of the original distribution.

(b) For all trees $T \in \operatorname{supp}(\boldsymbol{\rho})$, the vector $\nu(T)$ consists of exact max-marginals for $p(\mathbf{x}; \theta(T))$; we refer to this condition as *tree consistency.*

Our goal is to iteratively adjust the elements of ν — and hence, implicitly, $\boldsymbol{\theta}$ as well — such that the $\boldsymbol{\rho}$-reparameterization condition *always* holds, and the tree consistency condition is achieved upon convergence. In particular, we provide algorithms such that any fixed point ν^* satisfies both conditions (a) and (b).

The ultimate goal is to use ν^* to obtain an MAP configuration for the target distribution $p(\mathbf{x}; \bar{\theta})$. The following condition turns out to be critical in determining whether or not ν^* is useful for this purpose:

Optimum specification: The pseudo-max-marginals $\{\nu_s^*, \nu_{st}^*\}$ satisfy the optimum specification (OS) criterion if there exists at least one configuration $\mathbf{x}^*$ that satisfies the local optimality conditions (37) for every vertex $s \in V$ and edge $(s,t) \in E$ on the graph with cycles.

Note that the OS criterion always holds for any set of *exact* max-marginals on any graph. For the *pseudo*-max-marginals updated by the message-passing algorithms, in contrast, the OS criterion is no longer guaranteed to hold, as we illustrate in Example 4 to follow.

In the sequel, we establish that when ν^* satisfies the OS criterion with respect to some configuration $\mathbf{x}^*$, then any such $\mathbf{x}^*$ must be MAP-optimal for the target distribution. In contrast, when the OS criterion is not satisfied, the pseudo-max-marginals $\{\nu_s^*, \nu_{st}^*\}$ do not specify a MAP-optimal configuration, as can be seen by a continuation of Example 3.

Example 4 (Failure of the OS criterion). Consider the parameter vector $\bar{\theta}$ defined in equation (30). Let the spanning tree distribution $\boldsymbol{\rho}$ place mass 1/3 on each of the three spanning trees associated with a 3-node single cycle. With this choice, the edge appearances probabilities are $\rho_{st} = 2/3$ for each edge. We now define a 2-vector $\log \nu_s^*$ of log pseudo-max-marginals associated with node s, as well as a 2×2 matrix $\log \nu_{st}^*$ of log pseudo-max-marginals associated with edge (s,t), in the following way:

$$\log \nu_s^* \; := \; \begin{bmatrix} 0 & 0 \end{bmatrix} \quad \forall \, s \in V, \qquad\qquad \log \nu_{st}^* \; := \; \frac{1}{(2/3)} \begin{bmatrix} 0 & -\beta \\ -\beta & 0 \end{bmatrix} \quad \forall \, (s,t) \in E. \tag{41}$$

For each of the three trees in $\text{supp}(\boldsymbol{\rho})$, the associated vector $\nu^*(T)$ of pseudo-max-marginals defines a tree-structured exponential parameter $\theta^*(T)$ as in equation (39). More specifically, we have

$$\theta^*_s(T) \;=\; \log \nu^*_s \quad \forall\, s \;\in V, \qquad\qquad \theta^*_{st}(T) \;=\; \begin{cases} \log \nu^*_{st} & \forall\, (s,t) \in E(T), \\ 0 & \text{otherwise} \end{cases}$$

With this definition, it is straightforward to verify that $\sum_T \frac{1}{3}\theta^*(T) = \bar{\theta}$, meaning that the $\boldsymbol{\rho}$-reparameterization condition holds. Moreover, for any $\beta \in \mathbb{R}$, the edgewise consistency condition $\max_{x'_t} \nu^*_{st}(x_s, x_t) = \kappa\; \nu^*_s(x_s)$ holds. Therefore, the pseudo-max-marginals are pairwise-consistent, so that by Lemma 3, they are tree-consistent for all three spanning trees.

Now suppose that $\beta > 0$. In this case, the pseudo-max-marginal vector ν^* does satisfy the OS criterion. Indeed, both configurations [0 0 0] and [1 1 1] achieve $\text{argmax}_{x_s} \nu^*_s(x_s)$ for all vertices $s \in V$, and $\text{argmax}_{x_s, x_t} \nu^*_{st}(x_s, x_t)$ for all edges $(s,t) \in E$. This finding is consistent with Example 3, where we demonstrated that both configurations are MAP-optimal for the original problem, and that the LP relaxation (28) is tight.

Conversely, suppose that $\beta < 0$. In this case, the requirement that $\mathbf{x}^*$ belong to the set $\text{argmax}_{x_s, x_t} \nu^*_{st}(x_s, x_t)$ for *all* three edges means that $x^*_s \neq x^*_t$ for all three pairs. Since this condition cannot be met, the pseudo-max-marginal fails the OS criterion for $\beta < 0$. Again, this is consistent with Example 3, where we found that for $\beta < 0$, the optimum of the LP relaxation (28) was attained only at a fractional vertex. $\diamondsuit$

5.2.2 Direct updating of pseudo-max-marginals

Our first algorithm is based on updating a collection of pseudo-max-marginals $\{\nu_s, \nu_{st}\}$ for a graph with cycles such that $\boldsymbol{\rho}$-reparameterization (condition (a)) holds at every iteration, and tree consistency (condition (b)) is satisfied upon convergence. At each iteration $n = 0, 1, 2, \ldots$, associated with each node $s \in V$ is a single node pseudo-max-marginal ν^n_s, and with each edge $(s,t) \in E$ is a joint pairwise pseudo-max-marginal ν^n_{st}. Suppose that for each tree T in the support of $\boldsymbol{\rho}$, we use these pseudo-max-marginals $\{\nu^n_s, \nu^n_{st}\}$ to define a tree-structured exponential parameter $\theta^n(T)$ via equation (35). More precisely, again using the functional notation as in equation (11), the tree-structured parameter $\theta^n(T)$ is defined in terms of (element-wise) logarithms of ν^n as follows:

$$\theta^n_s(T)(x_s) \;=\; \log \nu^n_s(x_s) \qquad \forall\;\; s \in V \tag{42a}$$

$$\theta^n_{st}(T)(x_s, x_t) \;=\; \begin{cases} \log \frac{\nu^n_{st}(x_s, x_t)}{\nu^n_s(x_s)\nu^n_t(x_t)} & \text{if } (s,t) \in E(T) \\ 0 & \text{otherwise} \end{cases} \tag{42b}$$

The general idea is to update the pseudo-max-marginals iteratively, in such a way that the $\boldsymbol{\rho}$-reparameterization condition is maintained, and the tree consistency condition is satisfied upon convergence. There are a number of ways in which such updates can be structured; here we distinguish two broad classes of strategies: tree-based updates, and parallel edge-based updates. *Tree-based updates* entail performing multiple iterations of updating on a fixed tree $T \in \text{supp}(\boldsymbol{\rho})$, updating only the the subcollection $\nu(T)$ of pseudo-max-marginals associated with vertices and edges in T until it is fully tree-consistent *for this* tree (i.e., so that the components of $\nu(T)$ are indeed max-marginals for the distribution $p(\mathbf{x}; \theta(T))$). However, by focusing on this one tree, we may be changing some of the ν_s and ν_{st} so that we do *not* have tree-consistency on one or more of the other trees $T' \in \text{supp}(\boldsymbol{\rho})$. Thus, the next step entails updating the pseudo-max-marginals $\nu(T')$ on one of the other trees and so on, until ultimately the full collection ν is consistent on every tree. In contrast, the *edge-based strategy* involves updating the pseudo-max-marginal ν_{st} on each edge, as well as the associated single node max-marginals ν_s and ν_t, in parallel. This edgewise strategy is motivated by Lemma 3, which guarantees that ν is consistent on every tree of the graph if and only if the edge consistency condition

$$\max_{x'_t \in \mathcal{X}_t} \nu_{st}(x_s, x'_t) \quad = \quad \kappa\, \nu_s(x_s) \tag{43}$$

holds for every edge (s,t) of the graph with cycles.

It should be noted that tree-based updates are computationally feasible only when the support of the spanning tree distribution $\boldsymbol{\rho}$ consists of a manageable number of trees. When applicable, however, there can be important practical benefits to tree-based updates, including more rapid convergence as well as the possibility of determining a MAP-optimal configuration *prior* to convergence. We describe tree-based updates and their properties in more detail in Appendix B.

Here we focus on edge-based updates, due their simplicity and close link to the ordinary max-product algorithm that will be explored in the following section. The edge-based reparameterization algorithm takes the following form:

Algorithm 1 (Edge-based reparameterization updates).

1. Initialize the pseudo-max-marginals $\{\nu^0_s, \nu^0_{st}\}$ in terms of the original exponential parameter vector as follows:

$$\nu^0_s(x_s) \quad = \quad \kappa\; \exp\big(\bar{\theta}_s(x_s)\big) \tag{44a}$$

$$\nu^0_{st}(x_s, x_t) \quad = \quad \kappa\; \exp\big(\frac{1}{\rho_{st}}\bar{\theta}_{st}(x_s, x_t) + \bar{\theta}_t(x_t) + \bar{\theta}_s(x_s)\big) \tag{44b}$$

2. For iterations $n = 0, 1, 2, \ldots$, update the pseudo-max-marginals as follows:

$$\nu^{n+1}_s(x_s) \quad = \quad \kappa\; \nu^n_s(x_s) \prod_{t \in \Gamma(s)} \Big[\frac{\max_{x'_t} \nu^n_{st}(x_s, x'_t)}{\nu^n_s(x_s)}\Big]^{\rho_{st}} \tag{45a}$$

$$\nu^{n+1}_{st}(x_s, x_t) \quad = \quad \kappa\; \frac{\nu^n_{st}(x_s, x_t)}{\max_{x'_t} \nu^n_{st}(x_s, x'_t) \;\; \max_{x'_s} \nu^n_{st}(x'_s, x_t)} \;\; \nu^{n+1}_s(x_s)\nu^{n+1}_t(x_t) \tag{45b}$$

Remarks: (a) Each scalar ρ_{st} appearing equations (44b) and (45a) is the edge appearance probability of edge (s,t) induced by the spanning tree distribution $\boldsymbol{\rho}$, as defined in Section 3.1.
(b) This edge-based reparameterization algorithm is actually closely related to a reparameterization form of the ordinary max-product algorithm [36]. In fact, if $\rho_{st} = 1$ for all edges $(s,t) \in E$, then the update (45) are exactly equivalent to a reparameterization form of the usual max-product updates. We will see this equivalence in an alternative form in our subsequent presentation of explicit message-passing updates.

The following lemmas summarize the key properties of the Algorithm 1. We begin by claiming that all iterates of this algorithm specify a $\boldsymbol{\rho}$-reparameterization:

Lemma 5 ($\boldsymbol{\rho}$-reparameterization). *At each iteration $n = 0, 1, 2, \ldots$, the collection of tree-structured parameter vectors $\boldsymbol{\theta}^n = \{\theta^n(T)\}$, as specified by the pseudo-max-marginals $\{\nu_s^n, \nu_{st}^n\}$ via equation* (42)*, satisfies the $\boldsymbol{\rho}$-reparameterization condition.*

Proof. Using the initialization (44) and equation (42), for each tree $T \in \mathrm{supp}(\boldsymbol{\rho})$, we have the relation $\theta_{st}^0(T) = \bar{\theta}_{st}/\rho_{st}$ for all edges $(s,t) \in E(T)$, and $\theta_s^0(T) = \bar{\theta}_s$ for all vertices $s \in V$. Thus we have $\sum_T \rho(T)\theta_s^0(T) = \bar{\theta}_s$ for all $s \in V$ and $\sum_T \rho(T)\theta_{st}^0(T) = \sum_{T \ni (s,t)} \rho(T)\frac{\bar{\theta}_{st}}{\rho_{st}} = \bar{\theta}_{st}$ for all $(s,t) \in E$, so that $\boldsymbol{\rho}$-reparameterization holds for $n = 0$. We now proceed inductively: supposing that it holds for iteration n, we prove that it also holds for iteration $n+1$. Using the update equation (45) and equation (42), we find that for all $\mathbf{x} \in \mathcal{X}$, the weighted sum $\sum_T \rho(T)\theta^{n+1}(T)(\mathbf{x})$ is equal to

$$\sum_T \rho(T)\Bigg[\sum_{s\in V}\Bigg\{\log \nu_s^n(x_s) + \sum_{t\in\Gamma(s)} \rho_{st}\log\frac{\max_{x_t'} \nu_{st}^n(x_s, x_t')}{\nu_s^n(x_s)}\Bigg\} + \sum_{(s,t)\in E(T)} \log\frac{\nu_{st}^n(x_s,x_t)}{\max\limits_{x_t'} \nu_{st}^n(x_s,x_t') \ \max\limits_{x_s'} \nu_{st}^n(x_s',x_t)}\Bigg].$$

Some algebraic re-arrangement leads to an equivalent expression (up to additive constants independent of $\mathbf{x}$):

$$\sum_{s\in V} \log \nu_s^n(x_s) + \sum_{(s,t)\in E} \rho_{st} \log \frac{\nu_{st}^n(x_s,x_t)}{\nu_s^n(x_s)\nu_t^n(x_t)},$$

which, using equation (42), is seen to be equal to $\sum_T \rho(T)\theta^n(T)(\mathbf{x})$. Thus, the statement follows by the induction hypothesis. $\square$

Next we characterize the fixed points of the updates in step 2:

Lemma 6. *Any fixed point ν^* of the updates* (45) *satisfies the tree consistency condition (b).*

Proof. At a fixed point, we can substitute $\nu^* = \nu^n = \nu^{n+1}$ at all places in the updates. Doing so in equation (45b) and cancelling out common terms leads to the relation

$$\frac{\nu_{st}^*(x_s)}{\max_{x_t'} \nu_{st}^*(x_s, x_t')} \ \frac{\nu_t^*(x_t)}{\max_{x_s'} \nu_{st}^*(x_s', x_t)} = \kappa \qquad \forall \ (x_s, x_t),$$

from which the edgewise consistency condition (43) follows for each edge $(s,t) \in E$. The tree consistency condition then follows from Lemma 3. $\square$

5.2.3 Message-passing updates

The reparameterization updates of Algorithm 1 can also be described in terms of explicit message-passing operations. In this formulation, the pseudo-max-marginals depend on the original exponential parameter vector $\bar{\theta}$, as well as a set of auxiliary quantities $M_{st}(\cdot)$ associated with the edges of G. For each edge $(s,t) \in E$, $M_{st} : \mathcal{X}_t \to \mathbb{R}_+$ is a function from the state space $\mathcal{X}_t$ to the positive reals. The function $M_{st}(\cdot)$ represents information that is relayed from node s to node t, so that we refer to it as a "message". The resulting algorithm is an alternative but equivalent implementation of the reparameterization updates of Algorithm 1.

More explicitly, let us define pseudo-max-marginals $\{\nu_s, \nu_{st}\}$ in terms of $\bar{\theta}$ and a given set of messages $M = \{M_{st}\}$ as follows:

$$\nu_s(x_s) \ := \ \exp\big(\bar{\theta}_s(x_s)\big) \prod_{v\in\Gamma(s)} \big[M_{vs}(x_s)\big]^{\rho_{vs}} \tag{46a}$$

$$\nu_{st}(x_s,x_t) \ := \ \exp\big(\frac{1}{\rho_{st}}\bar{\theta}_{st}(x_s,x_t) + \bar{\theta}_s(x_s) + \bar{\theta}_t(x_t)\big) \ \frac{\prod_{v\in\Gamma(s)\backslash t}\big[M_{vs}(x_s)\big]^{\rho_{vs}}}{\big[M_{ts}(x_s)\big]^{(1-\rho_{ts})}} \ \frac{\prod_{v\in\Gamma(t)\backslash s}\big[M_{vt}(x_t)\big]^{\rho_{vt}}}{\big[M_{st}(x_t)\big]^{(1-\rho_{st})}} \tag{46b}$$

As before, these pseudo-max-marginals can be used to define a collection of tree-structured exponential parameters $\boldsymbol{\theta} = \{\theta(T)\}$ via equation (42). First, we claim that for any choice of messages, the set of tree-structured parameters so defined specifies a $\boldsymbol{\rho}$-reparameterization:

Lemma 7. *For any choice of messages, the collection $\{\theta(T)\}$ is a $\boldsymbol{\rho}$-reparameterization of $\bar{\theta}$.*

Proof. We use the definition (42) of $\theta(T)$ in terms of $\{\nu_s, \nu_{st}\}$ to write:

$$\begin{aligned}\sum_T \rho(T)\theta(T)(\mathbf{x}) &= \sum_T \rho(T)\Big[\sum_{s\in V}\log\nu_s(x_s) + \sum_{(s,t)\in E(T)} \log\frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)\nu_t(x_t)}\Big] \\ &= \sum_{s\in V}\log\nu_s(x_s) + \sum_{(s,t)\in E}\rho_{st}\log\frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)\nu_t(x_t)}, \qquad (47)\end{aligned}$$

where we have expanded out the expectation over $\boldsymbol{\rho}$ in the second line. Using the definition (46) of ν_s and ν_{st}, we have

$$\rho_{st}\log\frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)\nu_t(x_t)} = \bar{\theta}_{st}(x_s,x_t) - \rho_{st}\log M_{st}(x_t) - \rho_{st}\log M_{ts}(x_s),$$

As a consequence, each weighted log message $\rho_{st}\log M_{ts}(x_s)$ appears twice in equation (47): once in the term $\log\nu_s(x_s)$ with a plus sign, and once in the term $\log\nu_{st}(x_s,x_t)/\nu_s(x_s)\nu_t(x_t)$ with a negative sign. Therefore, the messages all cancel in the summation. This establishes that for all $\mathbf{x}\in\mathcal{X}^n$, we have $\sum_T \rho(T)\theta(T)(\mathbf{x}) = \sum_{s\in V}\bar{\theta}_s(x_s) + \sum_{(s,t)\in E}\bar{\theta}_{st}(x_s,x_t)$. □

The following set of message-passing updates are designed to find a collection of pseudo-max-marginals $\{\nu_s, \nu_{st}\}$ that satisfy the tree consistency condition (b).

Algorithm 2 (Parallel tree-reweighted max-product).

1. Initialize the messages $M^0 = \{M^0_{st}\}$ with arbitrary positive real numbers.

2. For iterations $n = 0, 1, 2, \ldots$, update the messages as follows:

$$M^{n+1}_{ts}(x_s) = \kappa \max_{x'_t\in\mathcal{X}_t}\Big\{\exp\Big(\frac{1}{\rho_{st}}\bar{\theta}_{st}(x_s,x'_t) + \bar{\theta}_t(x'_t)\Big)\frac{\prod_{v\in\Gamma(t)\backslash s}\big[M^n_{vt}(x'_t)\big]^{\rho_{vt}}}{\big[M^n_{st}(x'_t)\big]^{(1-\rho_{st})}}\Big\} \qquad (48)$$

Remarks: (a) The message-passing updates of Algorithm 2 represent an alternative way in which to implement the reparameterization updates of Algorithm 1. See Wainwright et al. [37] for further details of this equivalence between message-passing and reparameterization in the context of the sum-product algorithm.
(b) The message update equation (48) is closely related to the standard [22, 36] max-product updates, which correspond to taking $\rho_{st} = 1$ for every edge. On one hand, if the graph G is actually a tree, any vector in the spanning tree polytope must necessarily satisfy $\rho_{st} = 1$ for every edge $(s,t)\in E$, so that Algorithm 2 reduces to the ordinary max-product update. However, if G has cycles, then it is impossible to have $\rho_{st} = 1$ for every edge $(s,t)\in E$, so that the updates in equation (48) differ from the ordinary max-product updates in three critical ways. First, the

exponential parameters $\bar{\theta}_{st}(x_s, x_t)$ are scaled by the (inverse of the) edge appearance probability $1/\rho_{st} \geq 1$. Secondly, for each neighbor $v \in \Gamma(t) \backslash s$, the incoming message M_{vt} is exponentiated by the corresponding edge appearance probability $\rho_{vt} \leq 1$. Lastly, the update of message M_{ts} — that is, from t to s along edge (s,t) — depends on the *reverse direction* message M_{st} from s to t along the same edge. Despite these features, the messages can still be updated in an asynchronous manner, as in ordinary max-product [22, 36]. The tree-reweighted updates are related but distinct from the attenuated max-product updates proposed by Frey and Koetter [23]. A feature common to both algorithms is the re-weighting of messages; however, unlike the tree-reweighted update (48), the attenuated max-product update in [23] of the message from t to s does not involve the message in the reverse direction (i.e., from s to t).

By construction, any fixed point of Algorithm 2 specifies a set of tree-consistent pseudo-max-marginals, as summarized in the following:

Lemma 8. *For any fixed point M^* of the updates* (48)*, the associated pseudo-max-marginals ν^* defined as in equation* (46a) *and* (46b) *satisfy the tree-consistency condition.*

Proof. By Lemma 3, it suffices to verify that the edge consistency condition (43) holds for all edges $(s,t) \in E$. Using the definition (46) of ν_s^* and ν_{st}^*, the edge consistency condition (43) is equivalent to:

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

$$\kappa \max_{x_t' \in \mathcal{X}_t} \Big\{ \exp\big(\bar{\theta}_{st}(x_s, x_t') + \bar{\theta}_s(x_s) + \bar{\theta}_t(x_t')\big) \frac{\prod_{v \in \Gamma(s) \backslash t} \big[M_{vs}(x_s)\big]^{\rho_{vs}}}{\big[M_{ts}(x_s)\big]^{(1-\rho_{ts})}} \frac{\prod_{v \in \Gamma(t) \backslash s} \big[M_{vt}(x_t')\big]^{\rho_{vt}}}{\big[M_{st}(x_t')\big]^{(1-\rho_{st})}} \Big\}.$$

Pulling all terms involving $M_{ts}(x_s)$ to the LHS, and canceling out all remaining common terms yields the message update equation (48). $\square$

5.3 Analysis of algorithms

We now consider various questions associated with Algorithms 1 and 2, including existence of fixed points, convergence of the updates, and the relation of fixed points to the LP relaxation (28). As noted previously, the two algorithms (reparameterization and message-passing) represent alternative implementations of the same updates, and hence are equivalent in terms of their fixed point and convergence properties. For the purposes of the analysis given here, we focus on the message-passing updates given in Algorithm 2.

With reference to the first question, in related work [36], we proved the existence of fixed points for the ordinary max-product algorithm when applied to any strictly positive distribution defined on an arbitrary graph. The same proof can be adapted to show that the message-update equation (48) has at least one fixed point M^* under these same conditions. We do not yet have sufficient conditions to guarantee convergence on graphs with cycles; however, in practice, we find that the message updates (48) converge if suitably damped. In particular, we apply damping in the logarithmic domain, so that messages are updated according to $\lambda \log M_{ts}^{new} + (1-\lambda) \log M_{ts}^{old}$, where M_{ts}^{new} is calculated in equation (48). Finally, the following theorem addresses the nature of the fixed points, and in particular provides sufficient conditions for Algorithm 2 to yield exact MAP estimates for the target distribution $p(\mathbf{x}; \bar{\theta})$:

Theorem 2 (Exact MAP). *Let M^* be a fixed point of Algorithm 2, and suppose that the associated pseudo-max-marginals ν^* satisfy the optimum specification (OS) criterion. Then the following statements hold:*

(a) Any configuration $\mathbf{x}^*$ satisfying the local optimality conditions in the OS criterion is a MAP configuration for $p(\mathbf{x}; \bar{\theta})$.

(b) Let $\lambda^* = \log M^*$ be the logarithm of the fixed point M^* taken element-wise. Then λ^* is an optimal solution to the dual of the LP relaxation (28) *of Theorem 1.*

Proof. (a) By Lemma 7, the pseudo-max-marginals ν^* specify a $\boldsymbol{\rho}$-reparameterization of $p(\mathbf{x}; \bar{\theta})$. Since the message vector M^* defining ν^* is a fixed point of the update equation (48), Lemma 8 guarantees that the tree consistency condition holds. By the optimum specification (OS) criterion, we can find a configuration $\mathbf{x}^*$ that is node and edgewise optimal for ν^*. By Lemma 4, the configuration $\mathbf{x}^*$ is optimal for every tree-structured distribution $p(\mathbf{x}; \theta^*(T))$. Thus, by Proposition 1, the configuration $\mathbf{x}^*$ is MAP-optimal for $p(\mathbf{x}; \bar{\theta})$. Using the $\boldsymbol{\rho}$-reparameterization condition and the OS criterion, the value of this optimum is given by:

$$\Phi_\infty(\bar{\theta}) \quad = \quad \sum_{s \in V} \max_{x_s} \log \nu_s^*(x_s) + \sum_{(s,t) \in E} \rho_{st} \, \log \frac{\max_{x_s, x_t} \nu_{st}^*(x_s, x_t)}{\max_{x_s} \nu_s^*(x_s) \;\; \max_{x_t} \nu_t^*(x_t)}. \tag{49}$$

(b) The proof involves showing that the vector $\lambda^* := \log M^*$, defined by the element-wise logarithm of the message fixed point, is an optimal solution to a particular Lagrangian dual reformulation of the LP relaxation (28). It is convenient to represent any pseudomarginal τ more compactly in the functional form

$$\tau_s(x_s) \; := \; \sum_{j \in \mathcal{X}_s} \tau_{s;j} \delta_j(x_s), \qquad \tau_{st}(x_s, x_t) \; := \; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \tau_{st;jk} \delta_j(x_s) \delta_k(x_t).$$

For each edge (s,t) and element $x_s \in \mathcal{X}_s$, define $C_{ts}(x_s) := \tau_s(x_s) - \sum_{x_t' \in \mathcal{X}_t} \tau_{st}(x_s, x_t')$, and let $\lambda_{ts}(x_s)$ be a Lagrange multiplier associated with the constraint $C_{ts}(x_s) = 0$. We then consider the Lagrangian

$$\mathcal{L}_{\bar{\theta}, \boldsymbol{\rho}}(\tau, \lambda) \quad := \quad \langle \tau, \, \bar{\theta} \rangle + \sum_{(s,t) \in E} \rho_{st} \Big[\sum_{x_s \in \mathcal{X}_s} \lambda_{ts}(x_s) C_{ts}(x_s) + \sum_{x_t \in \mathcal{X}_t} \lambda_{st}(x_t) C_{st}(x_t) \Big]. \tag{50}$$

We have rescaled the Lagrange multipliers by the edge appearance probabilities $\rho_{st} > 0$ so as to make the connection to the messages in Algorithm 2 as explicit as possible. Indeed, let ν_s and ν_{st} represent pseudo-max-marginals, defined as in equation (46), with the message $M := \exp(\lambda)$. (Here the exponential is taken element-wise.) With the pseudo-max-marginals defined in this way, the following lemma provides a more convenient form of the Lagrangian:

Lemma 9. *Let r be an arbitrary node of the graph, and suppose that every tree $T \in \operatorname{supp}(\boldsymbol{\rho})$ is rooted at r, and the remaining edges are directed from parent-to-child ($s \to t$). Then an equivalent form of the Lagrangian* (50) *is $\mathcal{L}_{\bar{\theta}, \boldsymbol{\rho}}(\tau, \lambda) = \sum_T \rho(T) \mathcal{F}_{\bar{\theta}; \boldsymbol{\rho}}(\tau, \nu; T)$, where*

$$\mathcal{F}_{\bar{\theta}; \boldsymbol{\rho}}(\tau, \nu; T) \quad := \quad \sum_{x_r} \tau_r(x_r) \log \nu_r(x_r) \; + \; \sum_{(s \to t) \in E(T)} \sum_{(x_s, x_t)} \tau_{st}(x_s, x_t) \log \frac{\nu_{st}(x_s, x_t)}{\nu_s(x_s)}. \tag{51}$$

Proof. See Appendix C.1. □

Although the direction of any given edge may vary from tree to tree, this difference is not important for our purposes. With this set-up, we define the dual function as the maximization $\mathcal{Q}_{\bar{\theta};\boldsymbol{\rho}}(\lambda) := \max_{\tau \in \mathcal{S}} \mathcal{L}_{\bar{\theta},\boldsymbol{\rho}}(\tau, \lambda)$, over the constraint set

$$\mathcal{S} := \{\tau \mid \tau \geq 0, \; \sum_{x_s} \tau_s(x_s) = 1, \; \sum_{x_s, x_t} \tau_{st}(x_s, x_t) = 1\}.$$

Let M^* be a fixed point of the update equation (48), such that the pseudo-max-marginals ν^* satisfy the OS criterion.

Lemma 10. *The dual function evaluated at $\lambda^* := \log M^*$ is equal to:*

$$\mathcal{Q}_{\bar{\theta};\boldsymbol{\rho}}(\lambda^*) \;\; = \;\; \sum_{s \in V} \log \max_{x_s} \nu^*_s(x_s) \; + \sum_{(s,t) \in E} \rho_{st} \log \frac{\max_{x_s, x_t} \nu^*_{st}(x_s, x_t)}{\max_{x_s} \nu^*_s(x_s) \max_{x_t} \nu^*_t(x_t)}.$$

Proof. See Appendix C.2. □

By comparison to equation (49), we see that the dual value $\mathcal{Q}_{\bar{\theta};\boldsymbol{\rho}}(\lambda^*)$ is equal to the optimal primal value, which implies that $\lambda^* \equiv \log M^*$ must be a dual-optimal solution [7]. □

For a general graph with cycles, we have not established that any fixed point of Algorithm 2 (i.e., one for which ν^* does not satisfy the OS criterion) necessarily specifies a dual-optimal solution of the LP relaxation. However, on a tree-structured graph, the tree-reweighted max-product updates reduce to the ordinary max-product (min-sum) updates, and any fixed point ν^* must satisfy the OS criterion. In this case, we can use Theorem 2 to obtain the following corollary:

Corollary 2 (Ordinary max-product). *For a tree-structured graph T, the ordinary max-product algorithm is an iterative method for solving the dual of the exact LP representation of the MAP problem:*

$$\max_{\mathbf{x} \in \mathcal{X}^n} \langle \bar{\theta}, \, \boldsymbol{\phi}(\mathbf{x}) \rangle \;\; = \;\; \max_{\tau \in \mathrm{LOCAL}(T)} \langle \bar{\theta}, \, \tau \rangle. \tag{52}$$

Proof. By Lemma 1, the MAP problem $\max_{\mathbf{x} \in \mathcal{X}^n} \langle \bar{\theta}, \, \boldsymbol{\phi}(\mathbf{x}) \rangle$ has the alternative LP representation as $\max_{\tau \in \mathrm{MARG}(T)} \langle \bar{\theta}, \, \tau \rangle$. By the junction tree theorem [14], we have $\mathrm{MARG}(T) = \mathrm{LOCAL}(T)$ in the tree case, so that equation (52) follows. For the case of a tree, the only valid choice of $\boldsymbol{\rho_e}$ is the vector of all ones, so the tree-reweighted updates must be equivalent to the ordinary max-product algorithm. The result then follows from Theorem 2. □

Remark: It is important to recall that the tree-reweighted max-product algorithm differs from the ordinary max-product algorithm for non-trees. Indeed, the ordinary max-product algorithm — when applied to a graph with cycles — does *not* satisfy a condition analogous to Theorem 2(a). More precisely, it is possible to construct fixed points of the ordinary max-product algorithm that satisfy the OS criterion, yet the associated configuration $\mathbf{x}^*$ is not MAP-optimal. In contrast, any configuration specified by the tree-reweighted max-product updates is *guaranteed* to be MAP-optimal.

As proved in Corollaries 1 and 2, the techniques given here are always exact for tree-structured graphs. For graphs with cycles, their performance — more specifically, whether or not an MAP configuration can be obtained —- depends on both the graph structure, and the strength of the parameter vector $\bar{\theta}$. It is thus of particular interest to note that we have found several applications in which the tree relaxation and iterative algorithms described here are quite useful. For instance, we have applied the tree-reweighted max-product algorithm to a distributed data association problem involving multiple targets and sensors [13]. For the class of problem considered, the tree-reweighted max-product algorithm converges, typically quite rapidly, to a provably MAP-optimal data association. We have also applied these methods to decoding turbo-like and low density parity check (LDPC) codes [18, 19]. In the context of decoding, the fractional vertices of the polytope LOCAL(G) have a very concrete interpretation as pseudocodewords [e.g., 21, 24, 27, 38]. The performance of LP decoding is superior to the ordinary max-product algorithm, and slightly worse or comparable to the sum-product algorithm. An advantage of the methods given here, in contrast to the more standard iterative algorithms, is that they either acknowledge a decoding failure, or they output an MAP codeword accompanied by a guarantee of correctness. It remains to explore further the range of problems for which the iterative algorithms and LP relaxations described here are suitable.

6 Discussion

In this paper, we demonstrated the utility of convex combinations of tree-structured distributions in upper bounding the value of the maximum a posteriori (MAP) configuration on a Markov random field (MRF) on a graph with cycles. A key property is that this upper bound is met with equality if and only if the collection of tree-structured distributions shares a common optimum. Moreover, when the upper bound is tight, then then an MAP configuration can be obtained for the original MRF on the graph with cycles simply by examining the optima of the tree-structured distributions. This observation motivated two approaches for attempting to obtain tight upper bounds, and hence MAP configurations. First of all, we proved that the Lagrangian dual of the problem is equivalent to a linear programming (LP) relaxation, wherein the marginal polytope associated with the original MRF is replaced with a looser constraint set formed by tree-based consistency conditions. Interestingly, this constraint set is equivalent to the constraint set in the Bethe variational formulation of the sum-product algorithm [39]; in fact, the LP relaxation itself can be obtained by taking a suitable limit of the "convexified" Bethe variational problem analyzed in our previous work [35, 34]. Second, we developed a family of *tree-reweighted max product* algorithms that reparameterize a collection of tree-structured distributions in terms of a common set of pseudo-max-marginals on the nodes and edges of the graph with cycles. When it is possible to find a configuration that is locally optimal with respect to every single node and edge pseudo-max-marginal, then the upper bound is tight, and the MAP configuration can be obtained. Under this condition, we proved that fixed points of these message-passing algorithms specify dual-optimal solutions to the LP relaxation. A corollary of this analysis is that the ordinary max-product algorithm, when applied to trees, is solving the dual of an exact LP formulation of the MAP estimation problem.

Finally, in cases in which the methods described here do not yield MAP configurations, it is natural to consider strengthening the relaxation by forming clusters of random variables, as in the Kikuchi approximations described by Yedidia et al. [39]. In the context of this paper, this avenue amounts to taking convex combinations of hypertrees, which (roughly speaking) correspond to trees defined on clusters of nodes.[7] Such convex combinations of hypertrees lead, in the dual reformulation, to a hierarchy of progressively tighter LP relaxations, ordered in terms of the size of clusters used to form the hypertrees. On the message-passing side, it is also possible to develop

[7]See Wainwright et al. [37] for further details on hypertrees and their properties.

hypertree-reweighted forms of generalizations of the max-product algorithm.

Acknowledgments: We thank Jon Feldman and David Karger for helpful discussions.

A Proof of Lemma 2

By definition, we have $\Phi_\infty(\theta(T)) := \max_{\mathbf{x}\in\mathcal{X}^n} \langle \theta(T),\, \boldsymbol{\phi}(\mathbf{x})\rangle$. We re-write this function in the following way:

$$\Phi_\infty(\theta(T)) \stackrel{(a)}{=} \max_{\tau\in\mathrm{LOCAL}(G)} \langle \theta(T),\, \tau\rangle \stackrel{(b)}{=} \max_{\tau\in\mathrm{LOCAL}(G;T)} \langle \theta(T),\, \tau\rangle$$

where equality (a) follows from Lemma 1, and equality (b) follows because $\theta(T)_\alpha = 0$ for all $\alpha \notin \mathcal{I}(T)$. In this way, we recognize $\Phi_\infty(\theta(T))$ as the *support function* of the set $\mathrm{LOCAL}(G;T)$. Hence, by a standard result in convex analysis [26], the conjugate dual is the indicator function of $\mathrm{LOCAL}(G;T)$, as specified in equation (30).

For the sake of self-containment, we provide an explicit proof here. If τ belongs to $\mathrm{LOCAL}(G;T)$, then $\langle \theta(T),\, \tau\rangle - \Phi_\infty(\theta(T)) \leq 0$ holds for all $\theta(T) \in \mathcal{E}(T)$, with equality for $\theta(T) = 0$. From this relation, we conclude that

$$\sup_{\theta(T)\in\mathcal{E}(T)} \langle \theta(T),\, \tau\rangle - \Phi_\infty(\theta(T)) = 0$$

whenever $\tau \in \mathrm{LOCAL}(G;T)$.

On the other hand, if $\tau \notin \mathrm{LOCAL}(G;T)$, then by the (strong) separating hyperplane theorem [26], there must exist some vector γ and constant β such that (i) $\langle \gamma,\, \mu\rangle \leq \beta$ for all $\mu \in \mathrm{LOCAL}(G;T)$; and (ii) $\langle \gamma,\, \tau\rangle > \beta$. Since conditions (i) and (ii) do not depend on elements γ_α with $\alpha \notin \mathcal{I}(T)$, we can take $\gamma \equiv \gamma(T) \in \mathcal{E}(T)$ without loss of generality. We then have

$$\langle \gamma(T),\, \tau\rangle - \Phi_\infty(\gamma(T)) \;\geq\; \langle \gamma(T),\, \tau\rangle - \beta \;>\; 0. \tag{53}$$

Note that conditions (i) and (ii) are preserved under scaling of both $\gamma(T)$ and β by a positive number, so that can send the quantity (53) to positive infinity. We thus conclude that

$$\sup_{\theta(T)\in\mathcal{E}(T)} \big\{\langle \theta(T),\, \tau\rangle - \Phi_\infty(\theta(T))\big\} \;=\; +\infty$$

whenever $\tau \notin \mathrm{LOCAL}(G;T)$. This completes the proof of the lemma.

B Tree-based updates

This appendix provides a detailed description of tree-based updates. In this scheme, each iteration involves multiple rounds of message-passing on each tree T in the support of $\boldsymbol{\rho}$. More specifically, the computational engine used within each iteration is the ordinary max-product algorithm, applied as an *exact* technique to compute max-marginals for each tree-structured distribution.

At any iteration n, we let $\theta^n(T)$ denote a set of exponential parameters for the tree T. To be clear, the notation $\theta^n(T)$ used in this appendix differs slightly from its use in the main text. In particular, unlike in the main text, we can have $\theta^n_\alpha(T) \neq \theta^n_\alpha(T')$ for distinct trees $T = T'$ at immediate iterations, although upon convergence this equality will hold. Each step of the algorithm will involve computing, for every tree $T \in \mathrm{supp}(\boldsymbol{\rho})$, the max-marginals $\nu^n(T)$, associated with the tree-structured distribution $p(\mathbf{x};\theta(T))$. (Once again, unlike the main text, we need not have $\nu^n_\alpha(T) = \nu^n_\alpha(T')$ for distinct trees $T = T'$.)

Overall, the tree-based updates take the following form:

Algorithm 3 (Tree-based updates).

1. For each spanning tree $T \in \mathrm{supp}(\boldsymbol{\rho})$, initialize $\theta^0(T)$ via

$$
\begin{aligned}
\theta^0_s(T) &= \bar{\theta}_s \;\; \forall \;\; s \in V, \\
\theta^0_{st}(T) &= \frac{1}{\rho_{st}}\bar{\theta}_\alpha \;\; \forall \;\; (s,t) \in E(T), \qquad\qquad \theta^0_{st}(T) \;=\; 0 \;\; \forall \;\; (s,t) \in E \backslash E(T).
\end{aligned}
$$

2. For iterations $n = 0, 1, 2, \ldots$, do the following:

   (a) For each tree $T \in \mathrm{supp}(\boldsymbol{\rho})$, apply the ordinary max-product algorithm to compute the max-marginals $\nu^n(T)$ corresponding to the tree-structured distribution $p(\mathbf{x};\theta^n(T))$.

   (b) Check if the tree distributions share a common optimizing configuration (i.e., if $\cap_T \mathrm{OPT}(\theta^n(T))$ is non-empty).

      (i) If yes, output any shared configuration and terminate.

      (ii) If not, check to see whether or not the following agreement condition holds:

$$
\begin{aligned}
\nu_s(T) &= \nu_s(T') \qquad \forall \, s \in V, \qquad \forall T, T' \in \mathrm{supp}(\boldsymbol{\rho}), && (54a) \\
\nu_{st}(T) &= \nu_{st}(T') \qquad \forall \, T, \; T' \; \in \; \mathrm{supp}(\boldsymbol{\rho}) \;\; \mathrm{s.\,t} \; (s,t) \in E(T) \cap E(T'). && (54b)
\end{aligned}
$$

      If this agreement of all max-marginals holds, then terminate. Otherwise, form a new exponential parameter $\widetilde{\theta}$ as follows:

$$
\begin{aligned}
\widetilde{\theta}^{n+1}_s &= \sum_T \rho(T) \log \nu^n_s(T) \;\; \forall \;\; s \in V && (55a) \\
\widetilde{\theta}^{n+1}_{st} &= \sum_{T \ni (s,t)} \rho(T) \log \frac{\nu^n_{st}(T)}{\nu^n_s(T)\nu^n_t(T)} \;\; \forall \;\; (s,t) \in E && (55b)
\end{aligned}
$$

      Define $\theta^{n+1}(T)$ on each tree $T \in \mathrm{supp}(\boldsymbol{\rho})$ as in Step 1 with $\bar{\theta} \equiv \widetilde{\theta}$, and proceed to Step 2(a).

Termination: Observe that there are two possible ways in which Algorithm 3 can terminate. On one hand, the algorithm stops if in Step 2(b)(i), a collection of tree-structured distributions is found that all share a common optimizing configuration. Herein lies the possibility of *finite termination*, since there is no need to wait until the values of the tree max-marginals $\nu^n(T)$ all agree for every tree. Otherwise, the algorithm terminates in Step 2(b)(ii) if the max-marginals for each tree all agree with one another, as stipulated by equation (54).

A key property of the updates in Algorithm 3 is that they satisfy the $\boldsymbol{\rho}$-reparameterization condition:

Lemma 11. *For any iteration n, the tree-structured parameters $\{\theta^n(T)\}$ of Algorithm 3 satisfy $\sum_T \rho(T)\theta^n(T) = \bar{\theta}$.*

Proof. The claim for $n = 0$ follows from directly the initialization in Step 1. In particular, we clearly have $\sum_T \rho(T)\theta^0_s(T) = \bar{\theta}_s$ for any node $s \in V$. For any edge $(s,t) \in E$, we compute:

$$\sum_T \rho(T)\theta^0_{st}(T) \;=\; \sum_T \rho(T)\big[\frac{1}{\rho_{st}}\bar{\theta}_{st}\big] \;=\; \bar{\theta}_{st}.$$

To establish the claim for $n+1$, we proceed by induction. By the claim just proved, it suffices to show that $\widetilde{\theta}$, as defined in equation (55), defines the same distribution as $\bar{\theta}$.

Using the definition (55), we have

$$\begin{aligned}
\langle \widetilde{\theta},\, \boldsymbol{\phi}(\mathbf{x})\rangle &= \sum_{s\in V}\sum_{x_s}\widetilde{\theta}_s(x_s) + \sum_{(s,t)\in E}\sum_{x_s,x_t}\widetilde{\theta}_{st}(x_s,x_t) \\
&= \sum_{s\in V}\sum_{x_s}\sum_{T}\rho(T)\log\nu^n_s(T)(x_s) + \sum_{(s,t)\in E}\sum_{x_s,x_t}\sum_{T\ni(s,t)}\rho(T)\log\frac{\nu^n_{st}(T)(x_s,x_t)}{\nu^n_s(T)(x_s)\nu^n_t(T)(x_t)} \\
&= \sum_T\rho(T)\Big\{\sum_{s\in V}\sum_{x_s}\log\nu^n_s(T)(x_s) + \sum_{(s,t)\in E(T)}\sum_{x_s,x_t}\log\frac{\nu^n_{st}(T)(x_s,x_t)}{\nu^n_s(T)(x_s)\nu^n_t(T)(x_t)}\Big\} \qquad (56)
\end{aligned}$$

Recall that for each tree T, the quantities $\nu^n(T)$ are the max-marginals associated with the distribution $p(\mathbf{x};\theta^(T))$. Using the fact (35) that the max-marginals specify a reparameterization of $p(\mathbf{x};\theta^n(T))$, each term within curly braces is simply equal (up to an additive constant independent of $\mathbf{x}$) to $\langle\theta^n(T),\, \boldsymbol{\phi}(\mathbf{x})\rangle$. Therefore, by the induction hypothesis, the RHS of equation (56) is equal to $\langle\bar{\theta},\, \boldsymbol{\phi}(\mathbf{x})\rangle$, so that the claim of the lemma follows. □

On the basis of Lemma 11, it is straightforward to prove the analog of Theorem 2(a) for Algorithm 3. More specifically, whenever it outputs a configuration, it must be an exact MAP configuration for the original problem $p(\mathbf{x};\bar{\theta})$.

C Proof of Theorem 2

C.1 Proof of Lemma 9

We begin by re-expressing the Lagrangian (50) as follows:

$$\begin{aligned}
\mathcal{L}_{\bar{\theta},\boldsymbol{\rho}}(\tau,\lambda) = \sum_T \rho(T)\,\Big\{&\sum_{s\in V}\sum_{x_s}\tau_s(x_s)\big[\bar{\theta}_s(x_s) + \sum_{t\in\Gamma(s)}\rho_{st}\lambda_{ts}(x_s)\big] \\
&+ \sum_{(s,t)\in E(T)}\Big[\sum_{x_s,x_t}\tau_{st}(x_s,x_t)\big[\frac{\bar{\theta}_{st}(x_s,x_t)}{\rho_{st}} - \lambda_{st}(x_t) - \lambda_{ts}(x_s)\big]\Big]\Big\}.
\end{aligned}$$

Let ν_s and ν_{st} be the pseudo-max-marginals defined by the messages $M := \exp(\lambda)$. Then we can write

$$\mathcal{L}_{\bar{\theta},\boldsymbol{\rho}}(\tau,\lambda) \;=\; \sum_T\rho(T)\underbrace{\Big\{\sum_{s\in V}\sum_{x_s}\tau_s(x_s)\log\nu_s(x_s) + \sum_{(s,t)\in E(T)}\sum_{(j,k)}\tau_{st}(x_s,x_t)\log\frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)\nu_t(x_t)}\Big\}}_{\mathcal{F}_{\bar{\theta};\boldsymbol{\rho}}(\tau,\nu;T)}$$

Consider the quantity $\mathcal{F}_{\bar{\theta};\boldsymbol{\rho}}(\tau,\nu;T)$ for a fixed tree T. Let T be rooted at some vertex $r \in V$, and direct all the edges from child to parent $s \to t$. Using the fact that $\sum_{x_t} \tau_{st}(x_s,x_t) = \tau_s(x_s)$ on the constraint set, we can re-write $\mathcal{F}_{\bar{\theta};\boldsymbol{\rho}}(\tau,\nu;T)$ as

$$\mathcal{F}_{\bar{\theta};\boldsymbol{\rho}}(\tau,\nu;T) \;=\; \sum_{x_r} \tau_r(x_r)\log\nu_r(x_r) \;+\; \sum_{(s\to t)\in E(T)} \sum_{x_s,x_t} \tau_{st}(x_s,x_t)\log\frac{\nu_{st}(x_s,x_t)}{\nu_s(x_s)},$$

which proves the claim of Lemma 9.

C.2 Proof of Lemma 10

Since by assumption the pseudo-max-marginals ν^* defined by $M^* = \exp(\lambda^*)$ (with the exponential defined element-wise) satisfy the optimum specification criterion, we can find a configuration $\mathbf{x}^*$ that satisfies the local optimality conditions (37) for every node and edge on the full graph G.

Since the pseudo-max-marginals ν^* are defined by a fixed point $M^* = \exp(\lambda^*)$ (with the exponential defined element-wise) of the update equation (48), the pseudo-max-marginals must be pairwise-consistent. More explicitly, for any edge (s,t) and $x_s \in \mathcal{X}_s$, the pairwise consistency condition $\max_{x_t} \nu^*_{st}(x_s,x_t) = \kappa_{st}\,\nu^*(x_s)$ holds, where κ_{st} is a positive constant independent of x_s. Using this fact, we can write

$$\max_{x_s,x_t}\log\frac{\nu^*_{st}(x_s,x_t)}{\nu^*_s(x_s)} \;=\; \max_{x_s}\log\frac{\max_{x_t}\nu^*_{st}(x_s,x_t)}{\nu^*_s(x_s)} \;=\; \log\kappa_{st}. \tag{57}$$

Moreover, since by assumption the pseudo-max-marginals ν^* satisfy the optimum specification criterion, we can find a configuration $\mathbf{x}^*$ that satisfies the local optimality conditions (37) for every node and edge on the full graph G. For this configuration, we have the equality $\max_{x_s}\log\nu^*_s(x_s) = \log\nu^*_s(x^*_s)$ for all $s \in V$, as well as

$$\log\frac{\nu^*_{st}(x^*_s,x^*_t)}{\nu^*_s(x^*_s)} \;=\; \log\frac{\max_{x_t}\nu^*_{st}(x^*_s,x_t)}{\nu^*_s(x^*_s)} \;=\; \log\kappa_{st} \;=\; \max_{x_s,x_t}\log\frac{\nu^*_{st}(x_s,x_t)}{\nu^*_s(x_s)}, \tag{58}$$

for all $(s,t)\in E$, where the final equality follows from equation (57).

In conjunction with Lemma 9, equation (58) allows us to write the dual function at λ^* as follows:

$$\begin{aligned}
\mathcal{Q}_{\bar{\theta};\boldsymbol{\rho}}(\lambda^*) &= \sum_T \rho(T)\left\{\log\nu_r(x^*_r) \;+\; \sum_{(s\to t)\in E(T)} \log\frac{\nu^*_{st}(x^*_s,x^*_t)}{\nu^*_s(x^*_s)}\right\} \\
&= \sum_T \rho(T)\left\{\sum_{s\in V}\log\nu^*_s(x^*_s) \;+\; \sum_{(s,t)\in E(T)} \log\frac{\nu^*_{st}(x^*_s,x^*_t)}{\nu_s(x^*_s)\nu_t(x^*_t)}\right\} \\
&= \sum_{s\in V}\log\max_{x_s}\nu^*_s(x_s) \;+\; \sum_{(s,t)\in E}\rho_{st}\;\log\frac{\max_{x_s,x_t}\nu^*_{st}(x_s,x_t)}{\max_{x_s}\nu^*_s(x_s)\max_{x_t}\nu^*_t(x_t)},
\end{aligned}$$

which completes the proof of the claim.

References

[1] S. M. Aji, G. Horn, R. J. McEliece, and M. Xu. Iterative min-sum decoding of tail-biting codes. In *Proc. IEEE Information Theory Workshop, Killarney Ireland*, pages 68–69, June 1998.

[2] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Trans. Info. Theory*, 46:325–343, March 2000.

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

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

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

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

[7] D. Bertsimas and J. Tsitsikilis. *Introduction to linear optimization.* Athena Scientific, Belmont, MA, 1997.

[8] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–279, 1986.

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

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

[11] E. Boros, Y. Crama, and P. L. Hammer. Upper bounds for quadratic 0-1 maximization. *Operations Research Letters*, 9:73–79, 1990.

[12] P. Brémaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues.* Springer, 1991.

[13] L. Chen, M. J. Wainwright, M. Cetin, and A. Willsky. Multitarget-multisensor data association using the tree-reweighted max-product algorithm. In *SPIE Aerosense Conference*, April 2003.

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

[15] A. P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36, 1992.

[16] M. Deza and M. Laurent. *Geometry of cuts and metric embeddings.* Springer-Verlag, New York, 1997.

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

[18] J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo codes and its relation to iterative approaches. In *Proc. Allerton Conf. Communication, Control and Computing*, October 2002.

[19] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode binary linear codes. Technical report, Laboratory for Computer Science, MIT, 2003.

[20] G. D. Forney, Jr., F. R. Kschischang, B. Marcus, and S. Tuncel. Iterative decoding of tail-biting trellises and connections with symbolic dynamics. In *Codes, systems and graphical models*, pages 239–264. Springer, 2001.

[21] G.D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznick. On the effective weights of pseudocodewords for codes defined on graphs with cycles. In *Codes, systems and graphical models*, pages 101–112. Springer, 2001.

[22] W. T. Freeman and Y. Weiss. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. Info. Theory*, 47:736–744, 2001.

[23] B. J. Frey and R. Koetter. Exact inference using the attenuated max-product algorithm. In *Advanced mean field methods: Theory and Practice*. MIT Press, 2000.

[24] B. J. Frey, R. Koetter, and A. Vardy. Signal-space characterization of iterative decoding. *IEEE Trans. Info. Theory*, 47:766–781, 2001.

[25] P.L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation, and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28:121–155, 1984.

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

[27] G. Horn. *Iterative decoding and pseudocodewords*. PhD thesis, California Institute of Technology, 1999.

[28] S. L. Lauritzen. *Graphical models*. Oxford University Press, Oxford, 1996.

[29] R.J. McEliece, D.J.C. McKay, and J.F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Jour. Sel. Communication*, 16(2):140–152, February 1998.

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

[31] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

[32] S. Verdú. *Multiuser detection*. Cambridge University Press, Cambridge, 1998.

[33] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Info. Theory*, IT-13:260–269, April 1967.

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

[35] 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.

[36] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the max-product algorithm and its generalizations. Appeared as LIDS Tech. report, P-2554 MIT; Available online at `http://www.eecs.berkeley/edu/~wainwrig`, July 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, May 2003.

[38] N. Wiberg. *Codes and decoding on general graphs.* PhD thesis, University of Linkoping, Sweden, 1996.

[39] 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.