Message-passing for graph-structured linear programs: Proximal methods and rounding schemes

Pradeep Ravikumar[†]
pradeepr@stat.berkeley.edu

Alekh Agarwal[♯]
alekh@eecs.berkeley.edu

Martin J. Wainwright[†,♯]
wainwrig@stat.berkeley.edu

Department of Statistics[†], and
Department of Electrical Engineering and Computer Sciences[♯]
University of California, Berkeley
Berkeley, CA 94720

October 11, 2008

Abstract

The problem of computing a maximum a posteriori (MAP) configuration is a central computational challenge associated with Markov random fields. A line of work has focused on "tree-based" linear programming (LP) relaxations for the MAP problem. This paper develops a family of super-linearly convergent algorithms for solving these LPs, based on proximal minimization schemes using Bregman divergences. As with standard message-passing on graphs, the algorithms are distributed and exploit the underlying graphical structure, and so scale well to large problems. Our algorithms have a double-loop character, with the outer loop corresponding to the proximal sequence, and an inner loop of cyclic Bregman divergences used to compute each proximal update. Different choices of the Bregman divergence lead to conceptually related but distinct LP-solving algorithms. We establish convergence guarantees for our algorithms, and illustrate their performance via some simulations. We also develop two classes of graph-structured rounding schemes, randomized and deterministic, for obtaining integral configurations from the LP solutions. Our deterministic rounding schemes use a "re-parameterization" property of our algorithms so that when the LP solution is integral, the MAP solution can be obtained even before the LP-solver converges to the optimum. We also propose a graph-structured randomized rounding scheme that applies to iterative LP solving algorithms in general. We analyze the performance of our rounding schemes, giving bounds on the number of iterations required, when the LP is integral, for the rounding schemes to obtain the MAP solution. These bounds are expressed in terms of the strength of the potential functions, and the energy gap, which measures how well the integral MAP solution is separated from other integral configurations. We also report simulations comparing these rounding schemes.

1 Introduction

A key computational challenge that arises in applications of discrete graphical models is to compute the most probable configuration(s), often referred to as the *maximum a posteriori* (MAP) problem. Although the MAP problem can be solved exactly in polynomial time on trees (and more generally, graphs with bounded treewidth) using the max-product algorithm, it is computationally challenging for general graphs. Indeed, the MAP problem for general discrete graphical models includes a large number of classical NP-complete problems as special cases, including MAX-CUT, independent set, and various satisfiability problems.

This intractability motivates the development and analysis of methods for obtaining approximate solutions, and there is a long history of approaches to the problem. One class of methods is based on simulated annealing (Geman and Geman, 1984), but the cooling schedules required for theoretical guarantees are often prohibitively slow. Besag (1986) proposed the iterated conditional modes algorithm, which performs a sequence of greedy local maximizations to approximate the MAP solution, but may be trapped by local maxima. Greig et al. (1989) observed that for binary problems with attractive pairwise interactions (the ferromagnetic Ising model in statistical physics terminology), the MAP configuration can be computed in polynomial-time by reduction to a max-flow problem. The ordinary max-product algorithm, a form of non-serial dynamic-programming (Bertele and Brioschi, 1972), computes the MAP configuration exactly for trees, and is also frequently applied to graphs with cycles. Despite some local optimality results (Freeman and Weiss, 2001; Wainwright et al., 2004), it has no general correctness guarantees for graph with cycles, and even worse, it can converge rapidly to non-MAP configurations (Wainwright et al., 2005), even for problems that are easily solved in polynomial time (e.g., ferromagnetic Ising models). For certain graphical models arising in computer vision, Boykov et al. (2001) proposed graph-cut based search algorithms that compute a local maximum over two classes of moves. A broad class of methods are based on the principle of convex relaxation, in which the discrete MAP problem is relaxed to a convex optimization problem over continuous variables. Examples of this convex relaxation problem include linear programming relaxations (Koval and Schlesinger, 1976; Chekuri et al., 2005; Wainwright et al., 2005), as well as quadratic, semidefinite and other conic programming relaxations (for instance, (Ravikumar and Lafferty, 2006; Kumar et al., 2006; Wainwright and Jordan, 2004)).

Among the family of conic programming relaxations, linear programming (LP) relaxation is the least expensive computationally, and also the best understood. The primary focus of this paper is a well-known LP relaxation of the MAP estimation problem for pairwise Markov random fields, one which has been independently proposed by several groups (Koval and Schlesinger, 1976; Chekuri et al., 2005; Wainwright et al., 2005). This LP relaxation is based on optimizing over a set of locally consistent pseudomarginals on edges and vertices of the graph. It is an exact method for any tree-structured graph, so that it can be viewed naturally as a tree-based LP relaxation.[1] The first connection between max-product message-passing and LP relaxation was made by Wainwright et al. (2005), who connected the tree-based LP relaxation to the class of tree-reweighted max-product (TRW-MP) algorithms, showing that TRW-MP fixed points satisfying a strong "tree agreement" condition specify optimal solutions to the LP relaxation.

For general graphs, this first-order LP relaxation could be solved—at least in principle—by various standard algorithms for linear programming, including the simplex and interior-point methods (Bertsimas and Tsitsikilis, 1997; Boyd and Vandenberghe, 2004). However, such generic methods fail to exploit the graph-structured nature of the LP, and hence do not scale favorably to large-scale problems (Yanover et al., 2006). A body of work has extended the connection between the LP relaxation and message-passing algorithms in various ways. Kolmogorov (2005) developed a serial form of TRW-MP updates with certain convergence guarantees; he also showed that there exist fixed points of the TRW-MP algorithm, not satisfying strong tree agreement, that do not correspond to optimal solutions of the LP. This issue has a geometric interpretation, related to the fact that co-ordinate ascent schemes

[1] In fact, this LP relaxation is the first in a hierarchy of relaxations, based on the treewidth of the graph (Wainwright et al., 2005).

(to which TRW-MP is closely related), need not converge to the global optima for convex programs that are not strictly convex, but can become trapped in corners. Kolmogorov and Wainwright (2005) showed that this trapping phenomena does not arise for graphical models with binary variables and pairwise interactions, so that TRW-MP fixed points are always LP optimalgua. Globerson and Jaakkola (2007) developed a related but different dual-ascent algorithm, which is guaranteed to converge but is not guaranteed to solve the LP. Weiss et al. (2007) established connections between convex forms of the sum-product algorithm, and exactness of reweighted max-product algorithms; Johnson et al. (2007) also proposed algorithms related to convex forms of sum-product. Various authors have connected the ordinary max-product algorithm to the LP relaxation for special classes of combinatorial problems, including matching (Bayati et al., 2005; Huang and Jebara, 2007; Bayati et al., 2007) and independent set (Sanghavi et al., 2007). For general problems, max-product does *not* solve the LP; Wainwright et al. (2005) describe a instance of the MIN-CUT problem on which max-product fails, even though LP relaxation is exact. Other authors (Feldman et al., 2002; Komodakis et al., 2007) have implemented subgradient methods which are guaranteed to solve the linear program, but such methods typically have sub-linear convergence rates (Bertsimas and Tsitsikilis, 1997).

This paper makes two contributions to this line of work. Our first contribution is to develop and analyze a class of message-passing algorithms with the following properties: their only fixed points are LP-optimal solutions, they are provably convergent with at least a geometric rate, and they have a distributed nature, respecting the graphical structure of the problem. All of the algorithms in this paper are based on the well-established idea of *proximal minimization*: instead of directly solving the original linear program itself, we solve a sequence of so-called proximal problems, with the property that the sequence of associated solutions is guaranteed to converge to the LP solution. We describe different classes of algorithms, based on different choices of the proximal function: quadratic, entropic, and tree-reweighted entropies. For all choices, we show how the intermediate proximal problems can be solved by forms of message-passing on the graph—similar to but distinct from the ordinary max-product or sum-product updates. An additional desirable feature, given the wide variety of lifting methods for further constraining LP relaxations (Wainwright and Jordan, 2003), is that new constraints can be incorporated in a relatively seamless manner, by introducing new messages to enforce them.

Our second contribution is to develop various types of rounding schemes that allow for early termination of LP-solving algorithms. There is a substantial body of past work (e.g., (Raghavan and Thompson, 1987)) on rounding fractional LP solutions so as to obtain an integral solutions with approximation guarantees. Our use of rounding is rather different: instead, we consider rounding schemes applied to problems for which the LP solution is integral, so that rounding would be unnecessary if the LP were solved to optimality. In this setting, the benefit of certain rounding procedures (in particular, those that we develop) is allowing an LP-solving algorithm to be terminated *before* it has solved the LP, while still returning the MAP configuration, either with a deterministic or high probability guarantee. Our deterministic rounding schemes apply to a class of algorithms which, like the proximal minimization algorithms that we propose, maintain a certain invariant of the original problem. We also propose and analyze a class of graph-structured randomized rounding procedures that apply to any algorithm that approaches the optimal LP solution from the interior of the relaxed polytope. We analyze these rounding schemes, and give finite bounds on the number of iterations required for the rounding schemes to obtain an integral MAP solution.

The remainder of this paper is organized as follows. We begin in Section 2 with background on Markov random fields, and the first-order LP relaxation. In Section 3, we introduce the notions of proximal minimization and Bregman divergences, then derive various of message-passing algorithms based on these notions, and finally discuss their convergence properties. Section 4 is devoted to the development and analysis of rounding schemes, both for our proximal schemes as well as other classes of LP-solving algorithms. We provide experimental results in Section 5, and conclude with a discussion in Section 6.

2 Background

We begin by introducing some background on Markov random fields, and the LP relaxations that are the focus of this paper. Given a discrete space $\mathcal{X} = \{0, 1, 2, \ldots, m-1\}$, let $X = (X_1, \ldots, X_N) \in \mathcal{X}^N$ denote a N-dimensional discrete random vector. We assume that its distribution $\mathbb{P}$ is a Markov random field, meaning that it factors according to the structure of an undirected graph $G = (V, E)$, with each variable X_s associated with one node $s \in V$, in the following way. Letting $\theta_s : \mathcal{X} \to \mathbb{R}$ and $\theta_{st} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be singleton and edgewise potential functions respectively, we assume that the distribution takes the form

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

The problem of *maximum a posteriori* (MAP) estimation is to compute a configuration with maximum probability—i.e., an element

$$x^* \quad \in \quad \arg\max_{x \in \mathcal{X}^N} \big\{\sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)\big\}, \tag{1}$$

where the arg max operator extracts the configurations that achieve the maximal value. This problem is an integer program, since it involves optimizing over the discrete space $\mathcal{X}^N$. For future reference, we note that the functions $\theta_s(\cdot)$ and $\theta_{st}(\cdot)$ can always be represented in the form

$$\theta_s(x_s) = \sum_{j \in \mathcal{X}} \theta_{s;j} \mathbb{I}[x_s = j] \tag{2a}$$

$$\theta_{st}(x_s, x_t) = \sum_{j,k \in \mathcal{X}} \theta_{st;jk} \mathbb{I}[x_s = j;\, x_t = k], \tag{2b}$$

where the m-vectors $\{\theta_{s;j}, j \in \mathcal{X}\}$ and $m \times m$ matrices $\{\theta_{st;jk},\ (j,k) \in \mathcal{X} \times \mathcal{X}\}$ parameterize the problem.

The first-order linear programming (LP) relaxation (Koval and Schlesinger, 1976; Chekuri et al., 2005; Wainwright et al., 2005) of this problem is based on a set of pseudomarginals μ_s and μ_{st}, associated with the nodes and vertices of the graph. These pseudomarginals are constrained to be non-negative, as well to normalize and be locally consistent in the following sense:

$$\sum_{x_s \in \mathcal{X}} \mu_s(x_s) \quad = \quad 1, \qquad \text{for all } s \in V \text{, and} \tag{3a}$$

$$\sum_{x_t \in \mathcal{X}} \mu_{st}(x_s, x_t) \quad = \quad \mu_s(x_s) \qquad \text{for all } (s,t) \in E,\ x_s \in \mathcal{X}. \tag{3b}$$

The polytope defined by the non-negativity constraints $\mu \geq 0$, the normalization constraints (3a) and the marginalization constraints (3b), is denoted by $\mathbb{L}(G)$. The LP relaxation is based on maximizing the linear function

$$\langle \theta, \, \mu \rangle \quad := \quad \sum_{s \in V} \sum_{x_s} \theta_s(x_s) \mu_s(x_s) + \sum_{(s,t) \in E} \sum_{x_s, x_t} \theta_{st}(x_s, x_t) \mu_{st}(x_s, x_t), \tag{4}$$

subject to the constraint $\mu \in \mathbb{L}(G)$.

In the sequel, we write the linear program (4) more compactly in the form $\max_{\mu \in \mathbb{L}(G)} \langle \theta, \, \mu \rangle$. By construction, this relaxation is guaranteed to be exact for any problem on a tree-structured graph (Wainwright et al., 2005), so that it can be viewed as a tree-based relaxation. The main goal of this paper is to develop efficient and distributed algorithms for solving this LP relaxation, as well as strengthenings based on additional constraints. For instance, one natural strengthening is by "lifting": view the pairwise MRF as a particular case of a more general MRF with higher order cliques, define higher-order pseudomarginals on these cliques, and use them to impose higher-order consistency constraints. This particular progression of tighter relaxations underlies the Bethe to Kikuchi (sum-product to generalized sum-product) hierarchy (Yedidia et al., 2005); see Wainwright and Jordan (2003) for further discussion of such LP hierarchies.

3 Proximal minimization schemes

We begin by defining the notion of a proximal minimization scheme, and various types of divergences (among them Bregman) that we use to define our proximal sequences. Instead of dealing with the maximization problem $\max_{\mu \in \mathbb{L}(G)} \langle \theta, \, \mu \rangle$, it is convenient to consider the equivalent minimization problem $\min_{\mu \in \mathbb{L}(G)} -\langle \theta, \, \mu \rangle$.

3.1 Proximal minimization

The class of methods that we develop are based on the notion of proximal minimization (Bertsekas and Tsitsiklis, 1997). Instead of attempting to solve the LP directly, we solve a sequence of problems of the form

$$\mu^{n+1} = \arg \min_{\mu \in \mathbb{L}(G)} \Big\{ -\langle \theta, \, \mu \rangle + \frac{1}{\omega^n} D_f(\mu \, \| \, \mu^n) \Big\}, \tag{5}$$

where for iteration numbers $n = 0, 1, 2, \ldots$, the vector μ^n denotes current iterate, the quantity ω^n is a positive weight, and D_f is a generalized distance function, known as the proximal function. (Note that we are using superscripts to represent the iteration number, *not* for the power operation.)

The purpose of introducing the proximal function is to convert the original LP—which is convex but not strictly so—into a strictly convex problem. The latter property is desirable for a number of reasons. First, for strictly convex programs, co-ordinate descent schemes are guaranteed to converge to the global optimum; note that they may become trapped for non-strictly convex problems, such as the piecewise linear surfaces that arise in linear programming. Moreover, the dual of a strictly convex problem is guaranteed to be differentiable (Bertsekas, 1995); a guarantee which need not hold for non-strictly convex problems. Note that differentiable dual functions can in general be solved more easily than non-differentiable dual functions. In the sequel, we show how for appropriately chosen generalized distances, the proximal sequence $\{\mu^n\}$ can be computed using message passing updates derived from cyclic projections.

We note that the proximal scheme (5) is similar to an annealing scheme, in that it involves perturbing the original cost function, with a choice of weights $\{\omega^n\}$. While the weights $\{\omega^n\}$ can be adjusted for faster convergence, they can also be set to a constant, unlike for standard annealing procedures in which the annealing weight is taken to 0. The reason is that $D_f(\mu \,\|\, \mu^{(n)})$, as a generalized distance, itself converges to zero as the algorithm approaches the optimum, thus providing an "adaptive" annealing. For appropriate choice of weights and proximal functions, these proximal minimization schemes converge to the LP optimum with at least geometric and possibly superlinear rates (Bertsekas and Tsitsiklis, 1997; Iusem and Teboulle, 1995).

In this paper, we focus primarily on proximal functions that are Bregman divergences (Censor and Zenios, 1997), a class that includes various well-known divergences, among them the squared ℓ_2-distance and the Kullback-Leibler divergence. We say that f is a Bregman function if it is continuously differentiable, strictly convex, and has bounded level sets. Any such function induces a *Bregman divergence* as follows:

$$D_f(\mu' \,\|\, \nu) \quad := \quad f(\mu') - f(\nu) - \langle \nabla f(\nu),\, \mu' - \nu \rangle \tag{6}$$

Figure 1 shows this graphically. This divergence satisfies $D_f(\mu' \,\|\, \nu) \geq 0$ with equality if and only if $\mu' = \nu$, but need not be symmetric or satisfy the triangle inequality, so it is only a generalized distance. We study the sequence $\{\mu^n\}$ of proximal iterates (5) for the following choices of divergences.

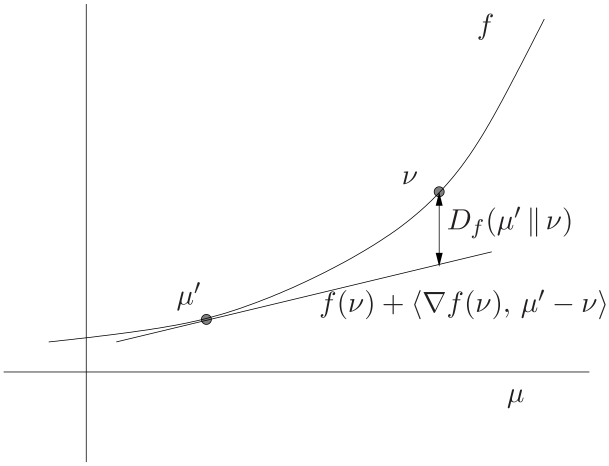

Figure 1: Graphical illustration of a Bregman divergence.

Quadratic divergence: This choice is the simplest, corresponding to the quadratic norm across nodes and edges

$$Q(\mu \,\|\, \nu) := \frac{1}{2}\sum_{s \in V} \|\mu_s - \nu_s\|^2 + \frac{1}{2}\sum_{(s,t)\in E} \|\mu_{st} - \nu_{st}\|^2, \tag{7}$$

where we have used the shorthand $\|\mu_s - \nu_s\|^2 = \sum_{x_s \in \mathcal{X}} |\mu_s(x_s) - \nu_s(x_s)|^2$, with similar notation for the edges. The underlying function that induces this Bregman divergence is simply the quadratic function

$$q(\mu) \quad := \quad \frac{1}{2}\Big\{ \sum_{s \in V} \sum_{x_s \in \mathcal{X}} \mu_s^2(x_s) + \sum_{(s,t) \in E} \sum_{(x_s, x_t) \in \mathcal{X} \times \mathcal{X}} \mu_{st}^2(x_s, x_t) \Big\}, \tag{8}$$

defined over nodes and edges of the graph.

Weighted entropic divergence: Another Bregman divergence can be defined by the weighted sum of Kullback-Leibler (KL) divergences across the nodes and edges. In particular, letting $\alpha_s > 0$ and $\alpha_{st} > 0$ be positive weights associated with node s and edge (s,t) respectively, we define

$$D_\alpha(\mu \,\|\, \nu) \quad = \quad \sum_{s \in V} \alpha_s D(\mu_s \,\|\, \nu_s) + \sum_{(s,t) \in E} \alpha_{st} D(\mu_{st} \,\|\, \nu_{st}), \tag{9}$$

where $D(p \,\|\, q) := \sum_x \big(p(x) \log \frac{p(x)}{q(x)} - \big[p(x) - q(x) \big] \big)$ is the KL divergence. An advantage of the KL divergence, relative to the quadratic norm, is that it automatically acts to enforce non-negativity constraints on the pseudomarginals in the proximal minimization problem. (See Section 3.4 for a more detailed discussion of this issue.) The associated Bregman function is weighted sum of entropies

$$h_\alpha(\mu) = \sum_{s \in V} \alpha_s H_s(\mu_s) + \sum_{(s,t) \in E} \alpha_{st} H_{st}(\mu_{st}), \tag{10}$$

where H_s and H_{st} are defined by

$$H_s(\mu_s) \; := \sum_{x_s \in \mathcal{X}} \mu_s(x_s) \log \mu_s(x_s), \qquad \text{and} \qquad H_{st}(\mu_{st}) \; := \sum_{(x_s, x_t) \in \mathcal{X} \times \mathcal{X}} \mu_{st}(x_s, x_t) \log \mu_{st}(x_s, x_t),$$

corresponding to node-based and edge-based entropies, respectively.

Tree-reweighted entropic divergence: Our last example is a divergence obtained from a convex combination of tree-structured entropies (Wainwright and Jordan, 2003). In particular, given a weight $\rho_{st} \in (0, 1]$ for each edge (s,t) of the graph, we define

$$f_\rho(\mu) \quad := \quad \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\mu_{st}). \tag{11}$$

In this definition, the quantity

$$I_{st}(\mu_{st}) \quad := \quad \sum_{(x_s, x_t) \in \mathcal{X} \times \mathcal{X}} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{[\sum_{x'_t} \mu_{st}(x_s, x'_t)][\sum_{x'_s} \mu_{st}(x'_s, x_t)]} \tag{12}$$

is the mutual information associated with edge (s,t). It can be shown that this function is strictly convex when restricted to $\mu \in \mathbb{L}(G)$. If $\mu \in \mathbb{L}(G)$, then the divergence induced by the function (11) is related to weighted entropy family (10), except that the node entropy weights α_s are not always positive.

3.2 Proximal sequences via Bregman projection

The key in designing an efficient proximal minimization scheme is ensuring that the proximal sequence $\{\mu^n\}$ can be computed efficiently. In this section, we first describe how sequences of proximal minimizations (when the proximal function is a Bregman divergence) can be reformulated as a particular Bregman projection. We then describe how this Bregman projection can itself be computed iteratively, in terms of a sequence of cyclic Bregman projections (Censor and Zenios, 1997) based on a decomposition of the constraint set $\mathbb{L}(G)$. In the sequel, we then show how this cyclic Bregman projections reduce to very simple message-passing updates.

Given a Bregman divergence D, the *Bregman projection* of the vector ν onto a convex set C is given by

$$\widehat{\mu} \quad := \quad \arg\min_{\mu \in C} D_f(\mu \,\|\, \nu) \tag{13}$$

By taking derivatives and using standard conditions for optima over convex sets (Bertsekas, 1995), the defining optimality condition for $\widehat{\mu}$ is

$$\langle \nabla f(\widehat{\mu}) - \nabla f(\nu),\, \mu - \widehat{\mu} \rangle \quad \geq \quad 0 \tag{14}$$

for all $\mu \in C$. Now consider the proximal minimization problem to be solved at step n, namely the strictly convex problem

$$\min_{\mu \in \mathbb{L}(G)} \Big\{ -\langle \theta,\, \mu \rangle + \frac{1}{\omega^n} D_f(\mu \,\|\, \mu^n) \Big\}. \tag{15}$$

By taking derivatives and using the same convex optimality, we see that the optimum μ^{n+1} is defined by the conditions

$$\langle \nabla f(\mu^{n+1}) - \nabla f(\mu^n) - \omega^n \theta,\, \mu - \mu^{n+1} \rangle \quad \geq \quad 0$$

for all $\mu \in C$. Note that these optimality conditions are of the same form as the Bregman projection conditions (14), with the vector $\nabla f(\mu^n) + \omega^n \theta$ taking the role of $\nabla f(\nu)$; in other words, with $(\nabla f)^{-1}(\nabla f(\mu) + \omega^n \theta)$ being substituted for ν. Consequently, efficient algorithms for computing the Bregman projection (14) can be leveraged to compute the proximal update (15). In particular, our algorithms leverage the fact that Bregman projections can be computed efficiently in a *cyclic manner*—that is, by decomposing the constraint set $C = \cap_i C_i$ into an intersection of simpler constraint sets, and then performing a sequence of projections onto these simple constraint sets (Censor and Zenios, 1997).

To simplify notation, for any Bregman function f, induced divergence D_f, and convex set C, let us define the operator $\mathrm{J}_f(\mu, \nu) := (\nabla f)^{-1}(\nabla f(\mu) + \nu)$, as well as the projection operator

$$\Pi_f(\gamma; C) := \arg\min_{\mu \in C} D_f(\mu \,\|\, \gamma).$$

With this notation, we can write the proximal update in a compact manner as the compounded operation

$$\mu^{n+1} \quad = \quad \Pi_f\Big(\mathrm{J}_f(\mu^n, \omega^n \theta); \mathbb{L}(G)\Big).$$

Now consider a decomposition of the constraint set as an intersection—say $\mathbb{L}(G) = \cap_{k=1}^{T} \mathbb{L}_k(G)$. By the method of cyclic Bregman projections (Censor and Zenios, 1997), we can compute

μ^{n+1} in an iterative manner, by performing the sequence of projections onto the simpler constraint sets, initializing $\mu^{n,0} = \mu^n$ and updating from $\mu^{n,\tau} \mapsto \mu^{n,\tau+1}$ by projecting $\mu^{n,\tau}$ onto constraint set $\mathbb{L}_{i(\tau)}(G)$, where $i(\tau) = \tau \mod T$, for instance. This meta-algorithm is summarized in Algorithm 1.

Algorithm 1 Basic proximal-Bregman LP solver

```
Given a Bregman distance $D$, weight sequence $\{\omega^n\}$ and problem parameters $\theta$:
  • Initialize $\mu^0$ to the uniform distribution: $\mu_s^{(0)}(x_s) = \frac{1}{m}$, $\mu_{st}^{(0)}(x_s, x_t) = \frac{1}{m^2}$.
  • Outer Loop: For iterations $n = 0, 1, 2, \ldots$, update $\mu^{n+1} = \Pi_f\Big(\mathrm{J}_f(\mu^n, \omega^n\theta); \mathbb{L}(G)\Big)$.
      – Solve Outer Loop via Inner Loop:
          (a) Inner initialization $\mu^{n,0} = \mathrm{J}_f(\mu^n, \omega^n\theta)$.
          (b) For $t = 0, 1, 2, \ldots$, set $i(t) = t \mod T$.
          (c) Update $\mu^{n,t+1} = \Pi_f\Big(\mu^{n,t}; \mathbb{L}_{i(t)}(G)\Big)$.
```

As shown in the following sections, by using a decomposition of $\mathbb{L}(G)$ over the edges of the graph, the inner loop steps correspond to local message-passing updates, slightly different in nature depending on the choice of Bregman distance. Iterating the inner and outer loops yields a provably convergent message-passing algorithm for the LP. Convergence follows from the convergence properties of proximal minimization (Bertsekas and Tsitsiklis, 1997), combined with convergence guarantees for cyclic Bregman projections (Censor and Zenios, 1997). In the following section, we derive the message-passing updates corresponding to various Bregman functions of interest.

3.3 Quadratic Projections

Consider the proximal sequence with the quadratic distance Q from equation (7); the Bregman function inducing this distance is the quadratic function $q(y) = \frac{1}{2}y^2$, with gradient $\nabla q(y) = y$. A little calculation shows that the operator J_q takes the form

$$\mathrm{J}_q(\mu, \omega\theta) \quad = \quad \mu + \omega\theta, \tag{16}$$

whence we obtain the initialization in equation (18a).

We now turn to the projections $\mu^{n,\tau+1} = \Pi_q(\mu^{n,\tau}, \mathbb{L}_i(G))$ onto the individual constraints $\mathbb{L}_i(G)$. For each such constraint, the local update is based on the solving the problem

$$\mu^{n,\tau+1} \quad = \quad \arg\min_{\nu \in \mathbb{L}_i(G)} \Big\{ q(\nu) - \langle \nu, \, \nabla q(\mu^{n,\tau}) \rangle \Big\}. \tag{17}$$

In Appendix A.1, we show how the solution to these inner updates takes the form (19a) given in Algorithm (2).

3.4 Entropic projections

Consider the proximal sequence with the Kullback-Leibler distance $D(\mu \,\|\, \nu)$ defined in equation (9). The Bregman function h_α inducing the distance is a sum of negative entropy functions

Algorithm 2 Quadratic Messages for μ^{n+1}

```
Initialization:
    $\mu_{st}^{(n,0)}(x_s, x_t) = \mu_{st}^{(n)}(x_s, x_t) + w^n \theta_{st}(x_s, x_t)$    (18a)
    $\mu_s^{(n,0)}(x_s) = \mu_s^{(n)}(x_s) + w^n \theta_s(x_s)$    (18b)
repeat
  for each edge $(s, t) \in E$ do
    $\mu_{st}^{(n,\tau+1)}(x_s, x_t) = \max\left\{\mu_{st}^{(n,\tau)}(x_s, x_t) + \left(\frac{1}{L+1}\right)\left(\mu_s^{(n,\tau)}(x_s) - \sum_{x_t} \mu_{st}^{(n,\tau)}(x_s, x_t)\right), 0\right\}$    (19a)
    $\mu_s^{(n,\tau+1)}(x_s) = \mu_s^{(n,\tau)}(x_s) + \left(\frac{1}{L+1}\right)\left(-\mu_s^{(n,\tau)}(x_s) + \sum_{x_t} \mu_{st}^{(n,\tau)}(x_s, x_t)\right)$    (19b)
  end for
  for each node $s \in V$ do
    $\mu_s^{(n,\tau+1)}(x_s) = \max\left\{0,\ \mu_s^{(n,\tau)}(x_s) + \frac{1}{L}\left(1 - \sum_{x_s} \mu_s^{(n,\tau)}(x_s)\right)\right\}$    (20)
  end for
until convergence
```

$f(\mu) = \mu \log \mu$, and its gradient is given by $\nabla f(\mu) = \log(\mu) + \vec{1}$. In this case, some calculation shows that the map $\nu = \mathrm{J}_f(\mu, \omega\theta)$ is given by

$$\nabla f(\nu) \;=\; \mu \exp(\omega\theta),$$

whence we obtain the initialization equation (21a). In Appendix A.2, we derive the message-passing updates summarized in Algorithm (3).
Remark: In the special case of uniformly weighted entropies (i.e., $\alpha_s = \alpha_{st} = 1$), it is worth noting that the updates of Algorithm (3) are reminiscent of the junction tree algorithm (Lauritzen, 1996), which also update the marginals $\{\mu_s, \mu_{st}\}$, or primal parameters directly.

3.5 Tree-reweighted entropy proximal sequences

In the previous sections, we saw how to solve the proximal sequences using message passing updates derived from cyclic Bregman projections. In this section, we show that for the tree-reweighted entropy divergence, we can also use tree-reweighted sum-product or related methods (Globerson and Jaakkola, 2007; Hazan and Shashua, 2008) to compute the proximal sequence. We first rewrite the proximal sequence optimization problem (5) as

$$\mu^{n+1} \;=\; \arg \min_{\nu \in \mathbb{L}(G)} \left\{ -\langle \omega\theta + \nabla f_\rho(\mu^n),\, \nu \rangle - f_\rho(\nu) \right\}. \tag{25}$$

Computing the gradient of the function f_ρ, and performing some algebra yields the algorithmic template of Algorithm 4. With this particular choice of proximal function, the resulting algorithm can be understood as approaching the zero-temperature limit of the tree-reweighted

Algorithm 3 Entropic Messages for μ^{n+1}

```
Initialization:
    $\mu_{st}^{(n,0)}(x_s, x_t) = \mu_{st}^{(n)}(x_s, x_t) \exp(\omega^n \theta_{st}(x_s, x_t)/\alpha_{st})$,   and   (21a)
    $\mu_s^{(n,0)}(x_s) = \mu_s^{(n)}(x_s) \exp(\omega^n \; \theta_s(x_s)/\alpha_s)$.   (21b)
repeat
    for each edge $(s,t) \in E$ do
        $\mu_{st}^{(n,\tau+1)}(x_s, x_t) = \mu_{st}^{(n,\tau)}(x_s, x_t) \left( \frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_t} \mu_{st}^{(n,\tau)}(x_s, x_t)} \right)^{\frac{\alpha_s}{\alpha_s + \alpha_{st}}}$,   and   (22)
        $\mu_s^{(n,\tau+1)}(x_s) = \mu_s^{(n,\tau)}(x_s)^{\frac{\alpha_s}{\alpha_s + \alpha_{st}}} \left( \sum_{x_t} \mu_{st}^{(n,\tau)}(x_s, x_t) \right)^{\frac{\alpha_{st}}{\alpha_s + \alpha_{st}}}$   (23)
    end for
    for each node $s \in V$ do
        $\mu_s^{(n,\tau+1)}(x_s) = \frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_s} \mu_s^{(n,\tau)}(x_s)}$   (24)
    end for
until convergence
```

Bethe problem; by convexity, the optimizers of this sequence are guaranteed to approach the LP optima (Wainwright and Jordan, 2003). Moreover, as pointed by Weiss et al. (2007), various other convexified entropies (in addition to the tree-reweighted one) also have this property.

3.6 Convergence

We now turn to the convergence of the message-passing algorithms that we have proposed. At a high-level, for any Bregman proximal function, convergence follows from two sets of known results: (a) convergence of proximal algorithms; and (b) convergence of cyclic Bregman projections.

For completeness, we re-state the consequences of these results here. For any positive sequence $\omega^n > 0$, we say that it satisfies the *infinite travel condition* if $\sum_{n=1}^{\infty}(1/\omega^n) = +\infty$. We let $\mu^* \in \mathbb{L}(G)$ denote an optimal solution (not necessarily unique) of the LP, and use $f^* = f(\mu^*) = \langle \theta, \, \mu^* \rangle$ to denote the LP optimal value. We say that the convergence rate is *superlinear* if

$$\lim_{n \to +\infty} \frac{|f(\mu^{n+1}) - f^*|}{|f(\mu^n) - f^*|} \;=\; 0, \tag{29}$$

and linear if

$$\lim_{n \to +\infty} \frac{|f(\mu^{n+1}) - f^*|}{|f(\mu^n) - f^*|} \;\leq\; \gamma, \tag{30}$$

Algorithm 4 TRW proximal solver

- For outer iterations $n = 0, 1, 2, \ldots,$

  (a) Update the parameters:

$$\widetilde{\theta}_s(x_s) = \omega^n \theta_s(x_s) + \log(\mu^n(x_s)) + 1 \tag{26}$$

$$\widetilde{\theta}_{st}(x_s, x_t) = \omega^n \theta_{st}(x_s, x_t) + \log \frac{\mu_{st}(x_s, x_t)}{\sum_{x_s'} \mu_{st}(x_s', x_t) \sum_{x_t'} \mu_{st}(x_s, x_t')} - 1 \tag{27}$$

  (b) Run a convergent TRW-solver to compute

$$\mu^{n+1} = \arg \min_{\nu \in \mathbb{L}(G)} \Big\{ - \langle \widetilde{\theta}^n, \, \nu \rangle - f_\rho(\nu) \Big\}. \tag{28}$$

for some $\gamma \in (0, 1)$.

Proposition 1 (Rate of outer loop convergence)**.** *Consider the sequence of iterates produced by a proximal algorithm* (5) *for LP-solving.*

(a) Using the quadratic proximal function and positive weight sequence $\omega^n \to 0$ satisfying infinite travel, the proximal sequence $\{\mu^n\}$ converges superlinearly.

(b) Using the entropic proximal function and positive weight sequence ω^n satisfying infinite travel, the proximal sequence $\{\mu^n\}$ converges:

(i) superlinearly if $\omega^n \to 0$, and

(ii) at least linearly if $\omega^n \geq c$ for some constant $c > 0$.

The quadratic case is covered in Bertsekas and Tsitsiklis (1997), whereas the entropic case was analyzed by Tseng and Bertsekas (1993), and Iusem and Teboulle (1995).

Our inner loop message updates use cyclic Bregman projections, for which there is also a substantial literature on convergence. Censor and Zenios (1997) show that with dual feasibility correction, cyclic projections onto general convex sets are convergent. For Euclidean projections with linear constraints, Deutsch et al. (2006) establish a geometric rate of convergence, dependent on angles between the half-spaces. The intuition is that the more orthogonal the half-spaces, the faster the convergence; for instance, a single iteration suffices for completely orthogonal constraints. Our inner updates thus converge geometrically to solution within each outer proximal step.

We note that the rate-of-convergence results for the outer proximal loops assume that the proximal update (computed within each inner loop) has been performed exactly. In practice, the solution to each proximal update might be computed only approximately, up to some accuracy ϵ. Some recent theory has addressed whether superlinear convergence can still be obtained in such a setting; for instance, Solodov (2001) shows that that under mild conditions superlinear rates still hold for proximal iterates with $\epsilon-$suboptimal inner-loop solutions. As we describe in Section 5, empirically, we have observed setting the termination threshold to $\epsilon = 10^{-6}$ is small enough to be practically irrelevant, in that superlinear convergence still occurs.

4 Rounding with optimality guarantees

Recall that the graph-structured LP (4) is a relaxation of the MAP integer program, so that there are two possible outcomes to solving the LP: either an integral vertex is obtained, which is then guaranteed to be a MAP configuration, or a fractional vertex is obtained, in which case the relaxation is loose. In the latter case, a natural strategy is to "round" the fractional solution, so as to obtain an integral solution (Raghavan and Thompson, 1987). Such rounding schemes may either be randomized or deterministic. A natural measure of the quality of the rounded solution is in terms of its value relative to the optimal (MAP) value. There is now a substantial literature on performance guarantees of various rounding schemes, when applied to particular sub-classes of MAP problems (e.g., (Raghavan and Thompson, 1987; Kleinberg and Tardos, 1999; Chekuri et al., 2005)).

In this section, we show that rounding schemes can be useful even when the LP optimum is integral, since they may permit an LP-solving algorithm to be *finitely terminated*—i.e., before it has actually solved the LP—while retaining the same optimality guarantees about the final output. An attractive feature of our proximal Bregman procedures is the existence of precisely such rounding schemes–namely, that under certain conditions, rounding pseudomarginals at intermediate iterations yields the integral LP optimum. In Section 4.1, we describe deterministic rounding schemes that apply to the proximal Bregman procedures that we have described; moreover, we provide upper bounds on the number of outer iterations required for the rounding scheme to obtain the LP optimum. In Section 4.2, we propose and analyze a graph-structured randomized rounding scheme, which applies not only to proximal Bregman procedures, but more broadly to any algorithm that generates a sequence of iterates contained within the local polytope $\mathbb{L}(G)$.

4.1 Deterministic rounding schemes

We begin by describing three deterministic rounding schemes that exploit the particular structure of the Bregman proximal updates.

Node-based rounding: This method is the simplest of the deterministic rounding procedures, and applies to the quadratic and entropic updates. It operates as follows: given the vector μ^n of pseudomarginals at iteration n, obtain an integral configuration $x^n(\mu^n) \in \mathcal{X}^N$ by choosing

$$x^n_s \quad \in \quad \arg\max_{x'_s \in \mathcal{X}} \mu^n(x'_s), \qquad \text{for each } s \in V. \tag{31}$$

We say that the node-rounded solution x^n is *edgewise-consistent* if

$$(x^n_s, x^n_t) \quad \in \quad \arg\max_{(x'_s, x'_t) \in \mathcal{X} \times \mathcal{X}} \mu^n_{st}(x'_s, x'_t) \qquad \text{for all edges } (s,t) \in E. \tag{32}$$

Neighborhood-based rounding: This rounding scheme applies to all three proximal schemes. For each node $s \in V$, denote its star-shaped neighborhood graph by $N_s = \{(s,t) | t \in N(s)\}$, consisting of edges between node s and its neighbors. Let $\{\text{QUA}, \text{ENT}, \text{TRW}\}$ refer to the quadratic, entropic, and tree-reweighted schemes respectively.

(a) Define the neighborhood-based energy function

$$F_s(x;\mu^n) \quad := \quad \begin{cases} 2\mu^n(x_s) + \sum\limits_{t\in N(s)} \mu^n(x_s,x_t) & \text{for QUA} \\ 2\alpha_s \log \mu^n_s(x_s) + \sum\limits_{t\in N(s)} \alpha_{st} \log \mu^n_{st}(x_s,x_t) & \text{for ENT} \\ 2\log \mu^n(x_s) + \sum\limits_{t\in N(s)} \rho_{st} \log \frac{\mu^n_{st}(x_s,x_t)}{\mu^n_s(x_s)\mu^n_t(x_t)} & \text{for TRW} \end{cases} \tag{33}$$

(b) Compute a configuration $x^n(N_s)$ maximizing the function $F_s(x;\mu^n)$ by running two rounds of ordinary max-product on the star graph.

Say that such a rounding is *neighborhood-consistent* if the neighborhood MAP solutions $\{x^n(N_s), s \in V\}$ agree on their overlaps.

Tree-based rounding: This method applies to all three proximal schemes, but most naturally to the TRW proximal method. Let $T_1,\ldots,T_K$ be a set of spanning trees that cover the graph (meaning that each edge appears in at least one tree), and let $\{\rho(T_i),\ i = 1,\ldots,K\}$ be a probability distribution over the trees. For each edge (s,t), define the *edge appearance probability* $\rho_{st} = \sum_{i=1}^K \rho(T_i)\ \mathbb{I}[(s,t) \in T_i]$. Then for each tree $i = 1,\ldots,K$:

(a) Define the tree-structured energy function

$$F_i(x;\mu^n) \quad := \quad \begin{cases} \sum\limits_{s\in V} \log \mu^n(x_s) + \sum\limits_{(s,t)\in E(T_i)} \frac{1}{\rho_{st}} \log \mu^n_{st}(x_s,x_t) & \text{for QUA} \\ \sum\limits_{s\in V} \alpha_s \log \mu^n(x_s) + \sum\limits_{(s,t)\in E(T_i)} \frac{\alpha_{st}}{\rho_{st}} \log \mu^n_{st}(x_s,x_t) & \text{for ENT} \\ \sum\limits_{s\in V} \log \mu^n(x_s) + \sum\limits_{(s,t)\in E(T_i)} \log \frac{\mu^n_{st}(x_s,x_t)}{\mu^n_s(x_s)\mu^n_t(x_t)} & \text{for TRW.} \end{cases} \tag{34}$$

(b) Run the ordinary max-product problem on energy $F_i(x;\mu^n)$ to find a MAP-optimal configuration $x^n(T_i)$.

Say that such a rounding is *tree-consistent* if the tree MAP solutions $\{x^n(T_i), i = 1,\ldots,M\}$ are all equal. This notion of tree-consistency is similar to the underlying motivation of the tree-reweighted max-product algorithm (Wainwright et al., 2005).

4.1.1 Optimality certificates for deterministic rounding

The following result characterizes the optimality guarantees associated with these rounding schemes, when they are consistent respectively in the *edge-consistency*, *neighborhood-consistency* and *tree-consistency* senses defined earlier.

Theorem 1 (Deterministic rounding with MAP certificate)**.** *Consider a sequence of iterates $\{\mu^n\}$ generated by the quadratic, entropic or TRW proximal schemes. For any $n = 1,2,3,\ldots$, any consistent rounded solution x^n obtained from μ^n via any of the node, neighborhood or tree-rounding schemes (when applicable) is guaranteed to be a MAP-optimal solution.*

We prove this claim in Section 4.1.2. It is important to note that such deterministic rounding guarantees do *not* apply to an arbitrary algorithm for solving the linear program. Rather, the algorithm must maintain a certain invariance that relates the rounded solution at an intermediate stage to the cost function of the original problem. These invariances are closely related to the reparameterization condition satisfied by the sum-product algorithm (Wainwright et al., 2003).

All of the rounding schemes require relatively little computation. The node-rounding scheme is trivial to implement. The neighborhood-based scheme requires running two iterations of max-product for each neighborhood of the graph. Finally, the tree-rounding scheme requires $\mathcal{O}(KN)$ iterations of max-product, where K is the number of trees that cover the graph, and N is the number of nodes. Many graphs with cycles can be covered with a small number K of trees; for instance, the lattice graph in 2-dimensions can be covered with two spanning trees, in which case the rounding cost is linear in the number of nodes.

Of course, the natural question is how many iterations n suffice for a given rounding scheme to succeed. The following result provides an upper bound on the number of iterations required for rounding success:

Corollary 1. *Suppose that the LP optimum is uniquely attained at an integral vertex μ^*, and that the sequence $\{\mu^n\}$ converges linearly—i.e., $|f(\mu^n) - f(\mu^*)| \leq |f(\mu^0) - f(\mu^*)|\gamma^n$ for some $\gamma \in (0, 1)$. Then there exists a constant C such for all iterations larger than $n^* := \frac{\log C|f(\mu^0)-f(\mu^*)|}{\log(1/\gamma)}$:*

(a) for quadratic and entropic schemes: all three types of rounding recover the MAP solution.

(b) for the TRW-based proximal method, tree-based rounding recovers the MAP solution.

Proof. We first claim that if the ℓ_∞-bound $\|\mu^n - \mu^*\|_\infty < \frac{1}{2}$ is satisfied, then the node-based rounding returns the (unique) MAP configuration, and moreover this MAP configuration x^* is edge-consistent with respect to μ^n. To see these facts, note that the ℓ_∞ bound implies, in particular, that at every node $s \in V$, we have

$$|\mu^n_s(x^*_s) - \mu^*_s(x^*_s)| \;=\; |\mu^n_s(x^*_s) - 1| \;<\; \frac{1}{2},$$

which implies that $\mu^n_s(x^*_s) > 1/2$ as $\mu^*_s(x^*_s) = 1$. Due to the non-negativity constraints and marginalization constraint $\sum_{x_s \in \mathcal{X}} \mu^n(x_s) = 1$, at most one configuration can have mass above $1/2$. Thus, node-based rounding returns x^*_s at each node s, and hence overall, it returns the MAP configuration x^*. The same argument also shows that the inequality $\mu^n_{st}(x^*_s, x^*_t) > \frac{1}{2}$ holds, which implies that $(x^*_s, x^*_t) = \arg\max_{x_s, x_t} \mu^n(x_s, x_t)$ for all $(s, t) \in E$. Thus, we have shown x^* is edge-consistent for μ^n_{st}, according to the definition (32).

We now bound the number of iterations required to achieve the ℓ_∞-bound. It suffices to show that $\|\mu^n - \mu^*\|_2 < 1/2$. There exists some constant $C > 0$ such that $\|\mu^n - \mu^*\|_2 \leq \frac{1}{2C}|f(\mu^n) - f(\mu^*)|$(cf. Prop. 8, Iusem and Teboulle (1995)). Consequently, we have

$$\|\mu^n - \mu^*\|_2 \;\;\leq\;\; \frac{|f(\mu^0) - f(\mu^*)|}{2C}\gamma^n.$$

Consequently, the choice of n^* given in the corollary statement shows that the ℓ_∞-bound is satisfied for all iterations $n \geq n^*$.

Next we turn to the performance of neighborhood rounding for the quadratic and entropic updates. For $n \geq n^*$, we know that x^* achieves the unique maximum of $\mu^n_s(x_s)$ at each node, and $\mu^n_{st}(x_s, x_t)$ on each edge. From the form of the neighborhood energy (33), this node- and edge-wise optimality implies that $x^*(N(s)) := \{x^*_t, t \in s \cup N(s)\}$ maximizes the neighborhood-based cost function as well, which implies success of neighborhood rounding. (Note that the positivity of the weights α_s and α_{st} is required to make this assertion.)

The proof of correctness for tree-rounding requires some elements from the proof of Theorem 1, so that we defer it to the end of Section 4.1.2. □

Note that we proved correctness of the neighborhood and tree-based rounding schemes by leveraging the correctness of the node-based rounding scheme. In practice, it is possible for neighborhood- or tree-based rounding to succeed even if node-based rounding fails; however, we currently do not have any sharper sufficient conditions for these rounding schemes.

4.1.2 Proof of Theorem 1

We now turn to the proof of Theorem 1. At a high level, the proof consists of two main steps. First, we show that each proximal algorithm maintains a certain invariant of the original MAP cost function $F(x;\theta)$; in particular, the iterate μ^n induces a reparameterization $F(x;\mu^n)$ of the cost function such that the set of maximizers is preserved—viz:

$$\arg\max_{x \in \mathcal{X}^N} F(x;\theta) := \arg\max_{x \in \mathcal{X}^N} \sum_{s \in V, x_s \in \mathcal{X}} \theta_s(x_s) + \sum_{(s,t) \in E, x_s, x_t \in \mathcal{X}} \theta_{st}(x_s, x_t) = \arg\max_{x \in \mathcal{X}^N} F(x;\mu^n). \tag{35}$$

Second, we show that the consistency conditions (edge, neighborhood or tree, respectively) guarantee that the rounded solution belongs to $\arg\max_{x \in \mathcal{X}^N} F(x;\mu^n)$

We begin with a lemma on the invariance property:

Lemma 1 (Invariance of maximizers)**.** *Define the function*

$$F(x;\mu) \quad := \begin{cases} \sum\limits_{s \in V} \mu_s(x_s) + \sum\limits_{(s,t) \in E} \mu_{st}(x_s, x_t) & \textit{for } \text{QUA} \\ \sum\limits_{s \in V} \alpha_s \log \mu_s(x_s) + \sum\limits_{(s,t) \in E} \alpha_{st} \log \mu_{st}(x_s, x_t) & \textit{for } \text{ENT} \\ \sum\limits_{s \in V} \log \mu_s(x_s) + \sum\limits_{(s,t) \in E} \rho_{st} \log \frac{\mu_{st}(x_s,x_t)}{\mu_s(x_s)\mu_t(x_t)} & \textit{for } \text{TRW} \end{cases} \tag{36}$$

At each iteration $n = 1, 2, 3, \ldots$ for which $\mu^n > 0$, the function $F(x;\mu^n)$ preserves the set of maximizers (35)*.*

The proof of this claim, provided in Appendix B, is based on exploiting the necessary (Lagrangian) conditions defined by the optimization problems characterizing the sequence of iterations $\{\mu^n\}$.

For the second part of the proof, we show that how a solution x^*, obtained by a rounding procedure, is guaranteed to maximize the function $F(x;\mu^n)$, and hence (by Lemma 1) the original cost function $F(x;\theta)$. In particular, we state the following simple lemma:

Lemma 2. *The rounding procedures have the following guarantees:*

(a) Any edge-consistent configuration from node rounding maximizes $F(x;\mu^n)$ for the quadratic and entropic schemes.

(b) Any neighborhood-consistent configuration from neighborhood rounding maximizes $F(x;\mu^n)$ for the quadratic and entropic schemes.

(c) Any tree-consistent configuration from tree rounding maximizes $F(x;\mu^n)$ for all three schemes.

Proof. We begin by proving statement (a). Consider an edge-consistent integral configuration x^* obtained from node rounding. By definition, it maximizes $\mu^n(x_s)$ for all $s \in V$, and $\mu^n_{st}(x_s, x_t)$ for all $(s,t) \in E$, and so by inspection, also maximizes $F(x; \mu^n)$ for the quadratic and proximal cases.

We next prove statement (b) on neighborhood rounding. Suppose that neighborhood rounding outputs a single neighborhood-consistent integral configuration x^*. Since $x^*_{N(s)}$ maximizes the neighborhood energy (33) at each node $s \in V$, it must also maximize the sum $\sum_{s \in V} F_s(x; \mu^n)$. A little calculation shows that this sum is equal to $2F(x; \mu^n)$, the factor of two arising since the term on edge (s,t) arises twice, one for neighborhood rooted at s, and once for t.

Turning to claim (c), let x^* be a tree-consistent configuration obtained from tree rounding. Then for each $i = 1, \ldots, K$, the configuration x^* maximizes the tree-structured function $F_i(x; \mu^n)$, and hence also maximizes the convex combination $\sum_{i=1}^K \rho(T_i) F_i(x; \mu^n)$. By definition of the edge appearance probabilities ρ_{st}, this convex combination is equal to the function $F(x; \mu^n)$. □

4.2 Randomized rounding

The schemes considered in the previous section were all deterministic, since (disregarding any possible ties), the output of the rounding procedure was a deterministic function of the given pseudomarginals $\{\mu^n_s, \mu^n_{st}\}$. In this section, we consider randomized rounding procedures, in which the output is a random variable.

Perhaps the most naive randomized rounding scheme is the following: for each node $r \in V$, assign it value $x_r \in \mathcal{X}$ with probability $\mu^n_v(x_r)$. We propose a graph-structured generalization of this naive randomized rounding scheme, in which we perform the rounding in a dependent way across sub-groups of nodes, and establish guarantees for its success. In particular, we show that when the LP relaxation has a unique integral optimum that is well-separated from the second best configuration, then the rounding scheme succeeds with high probability after a pre-specified number of iterations.

4.2.1 The randomized rounding scheme

Our randomized rounding scheme is based on any given subset E' of the edge set E. Consider the subgraph $G(E \backslash E')$, with vertex set V, and edge set $E \backslash E'$. We assume that E' is chosen such that the subgraph $G(E \backslash E')$ is a forest. That is, we can decompose $G(E \backslash E')$ into a union of disjoint trees, $\{T_1, \ldots, T_K\}$, where $T_i = (V_i, E_i)$, such that the vertex subsets V_i are all disjoint and $V = V_1 \cup V_2 \cup \ldots \cup V_K$. We refer to the edge subset as *forest-inducing* when it has this property. Note that such a subset always exists, since $E' = E$ is trivially forest-inducing. In this case, the "trees" simply correspond to individual nodes, without any edges; $V_i = \{i\}$, $E_i = \emptyset$, $i = 1, \ldots, N$.

For any forest-inducing subset $E' \subseteq E$, Algorithm 5 defines our randomized rounding scheme.

To be clear, the randomized solution X is a function of both the pseudomarginals μ, and the choice of forest-inducing subset E', so that we occasionally use the notation $X(\mu; E')$ to reflect explicitly this dependence. Note that the simplest rounding scheme of this type is obtained by setting $E' = E$. Then the "trees" simply correspond to individual nodes without any edges, and the rounding scheme is the trivial node-based scheme.

Algorithm 5 Randomized rounding scheme

```
for subtree indices $i = 1, \ldots, K$ do
    Sample a sub-configuration $X_{V_i}$ from the probability distribution
        $p(x_{V_i}; \mu(T_i)) = \prod_{s \in V_i} \mu^n(x_s) \prod_{(s,t) \in E_i} \frac{\mu(x_s, x_t)}{\mu(x_s)\mu(x_t)}$.    (37)
end for
Form the global configuration $X \in \mathcal{X}^N$ by concatenating all the local random samples:
    $X := \Big(X_{V_1}, \ldots, X_{V_K}\Big)$.
```

The randomized rounding scheme can be "derandomized" so that we obtain a deterministic solution $x^d(\mu^n; E')$ that does at least well as the randomized scheme does in expectation. This derandomization scheme is shown in Algorithm 6, and its correctness is guaranteed in the following theorem, proved in Appendix C.

Theorem 2. *Let $(G = (V, E),\ \theta)$ be the given MAP problem instance, and let $\mu^n \in \mathbb{L}(G)$ be any set of pseudomarginals in the local polytope $\mathbb{L}(G)$. Then, for any subset $E' \subseteq E$ of the graph G, the (E', μ^n)-randomized rounding scheme in Algorithm 5, when derandomized as in Algorithm 6 satisfies,*

$$F(x^d(\mu^n; E'); \theta) \geq \mathbb{E}\Big(F(X(\mu^n; E'); \theta)\Big),$$

where $X(\mu^n; E')$ and $x^d(\mu^n; E')$ denote the outputs of the randomized and derandomized schemes respectively.

4.2.2 Oscillation and gaps

In order to state some theoretical guarantees on our randomized rounding schemes, we require some notation. For any edge $(s, t) \in E$, we define the *edge-based oscillation*

$$\delta_{st}(\theta) := \max_{x_s, x_t} [\theta_{st}(x_s, x_t)] - \min_{x_s, x_t} [\theta_{st}(x_s, x_t)] \tag{38}$$

We define the *node-based oscillation* $\delta_s(\theta)$ in the analogous manner. The quantities $\delta_s(\theta)$ and $\delta_{st}(\theta)$ are measures of the strength of the potential functions.

We extend these measures of interaction strength to the full graph in the natural way

$$\delta_G(\theta) := \max\Big\{ \max_{(s,t) \in E} \delta_{st}(\theta),\ \max_{s \in V} \delta_s(\theta) \Big\}. \tag{39}$$

Using this oscillation function, we now define a measure of the quality of a unique MAP optimum, based on its separation from the second most probable configuration. In particular, letting $x^* \in \mathcal{X}^N$ denote a MAP configuration, and recalling the notation $F(x; \theta)$ for the LP objective, we define the *graph-based gap*

$$\Delta(\theta; G) := \frac{\min_{x \neq x^*} \Big[F(x^*; \theta) - F(x; \theta) \Big]}{\delta_G(\theta)}. \tag{40}$$

Algorithm 6 Derandomized rounding scheme

```
Initialize: $\bar{\mu} = \mu^n$.

for subtree indices $i = 1, \ldots, K$ do
  Solve
    $x^d_{V_i} = \arg\max_{x_{V_i}} \sum_{s \in V_i} \Big\{ \theta_s(x_s) + \sum_{t\colon (s,t) \in E'} \sum_{x_t} \bar{\mu}_t(x_t) \theta_{st}(x_s, x_t) \Big\} + \sum_{(s,t) \in E_i} \theta_{st}(x_s, x_t).$
  Update $\bar{\mu}$:
    $\bar{\mu}_s(x_s) = \begin{cases} \bar{\mu}_s(x_s) & \text{if } s \notin V_i \\ 0 & \text{if } s \in V_i, x^d_s \neq x_s \\ 1 & \text{if } s \in V_i, x^d_s = x_s \end{cases}$
    $\bar{\mu}_{st}(x_s, x_t) = \begin{cases} \bar{\mu}_{st}(x_s, x_t) & \text{if } (s,t) \notin E_i \\ \bar{\mu}(x_s)\bar{\mu}_t(x_t) & \text{if } (s,t) \in E_i \end{cases}$
end for
Form the global configuration $x^d \in \mathcal{X}^N$ by concatenating all the subtree configurations:
  $x^d := \Big( x^d_{V_1}, \ldots, x^d_{V_K} \Big).$
```

This gap function is a measure of how well-separated the MAP optimum x^* is from the remaining integral configurations. By definition, the gap $\Delta(\theta; G)$ is always non-negative, and it is strictly positive whenever the MAP configuration x^* is unique. Finally, note that the gap is invariant to the translations ($\theta \mapsto \theta' = \theta + C$) and rescalings ($\theta \mapsto \theta' = c\theta$) of the parameter vector θ. These invariances are appropriate for the MAP problem since the optima of the energy function $F(x; \theta)$ are not affected by either transformation (i.e., $\arg\max_x F(x; \theta) = \arg\max_x F(x; \theta')$ for both $\theta' = \theta + C$ and $\theta' = c\theta$).

Finally, for any forest-inducing subset, we let $d(E')$ be the maximum degree of any node with respect to edges in E'—namely,

$$d(E') \;:=\; \max_{s \in V} |t \in V \;\mid\; (s,t) \in E'|.$$

4.2.3 Optimality Guarantees

We show, in this section, that when the pseudomarginals μ^n are within a specified ℓ_1 norm ball around the unique MAP optimum μ^*, the randomized rounding scheme outputs the MAP configuration with high probability.

Theorem 3. *Consider a problem instance (G, θ) for which the MAP optimum x^* is unique, and let μ^* be the associated vertex of the polytope $\mathbb{L}(G)$. For any $\epsilon \in (0,1)$, if at some iteration n, we have $\mu^n \in \mathbb{L}(G)$, and*

$$\|\mu^n - \mu^*\|_1 \;\leq\; \frac{\epsilon\, \Delta(\theta; G)}{1 + d(E')}, \tag{41}$$

then (E', μ^n)-randomized rounding succeeds with probability greater than $1 - \epsilon$,

$$\mathbb{P}[X(\mu^n; E') = x^*] \;\;\geq\;\; 1 - \epsilon$$

We provide the proof of this claim in Appendix D. It is worthwhile observing that the theorem applies to any algorithm that generates a sequence $\{\mu^n\}$ of iterates contained within the local polytope $\mathbb{L}(G)$. In addition to the proximal Bregman updates discussed in this paper, it also applies to interior-point methods (Boyd and Vandenberghe, 2004) for solving LPs. For the naive rounding based on $E' = E$, the sequence $\{\mu^n\}$ need not belong to $\mathbb{L}(G)$, but instead need only satisfy the milder conditions $\mu^n_s(x_s) \geq 0$ for all $s \in V$ and $x_s \in \mathcal{X}$, and $\sum_{x_s} \mu^n_s(x_s) = 1$ for all $s \in V$.

The derandomized rounding scheme enjoys a similar guarantee, as shown in the following theorem, proved in Appendix E.

Theorem 4. *Consider a problem instance (G, θ) for which the MAP optimum x^* is unique, and let μ^* be the associated vertex of the polytope $\mathbb{L}(G)$. If at some iteration n, we have $\mu^n \in \mathbb{L}(G)$, and*

$$\|\mu^n - \mu^*\|_1 \;\leq\; \frac{\Delta(\theta; G)}{1 + d(E')},$$

then the (E', μ^n)-derandomized rounding scheme in Algorithm 6 outputs the MAP solution,

$$x^d(\mu^n; E') = x^*.$$

4.2.4 Bounds on iterations

Although Theorems 3 and 4 apply even for sequences $\{\mu^n\}$ that need not converge to μ^*, it is most interesting when the LP relaxation is tight, so that the sequence $\{\mu^n\}$ generated by any LP-solver satisfies the condition $\mu^n \to \mu^*$. In this case, we are guaranteed that for any fixed $\epsilon \in (0, 1)$, the bound (41) will hold for an iteration number n that is "large enough". Of course, making this intuition precise requires control of convergence rates. Recall that N is the number of nodes in the graph, and m is cardinality of the set $\mathcal{X}$ from which all variables takes their values. With this notation, we have the following.

Corollary 2. *Under the conditions of Theorem 3, suppose that the sequence of iterates $\{\mu^n\}$ converge to the LP (and MAP) optimum at a linear rate: $\|\mu^n - \mu^*\|_2 \leq \gamma^n \|\mu^0 - \mu^*\|_2$. Then:*

(a) The randomized rounding in Algorithm 5 succeeds with probability at least $1 - \epsilon$ for all iterations greater than

$$n^* \;:=\; \frac{\frac{1}{2}\log\left(Nm + N^2m^2\right) \,+ \log\left(\|\mu^0 - \mu^*\|_2\right) \,+ \log\left(\frac{1+d(E')}{\Delta(\theta;G)}\right) + \log(1/\epsilon)}{\log(1/\gamma)}.$$

(b) The derandomized rounding in Algorithm 6 yields the MAP solution for all iterations greater than

$$n^* \;:=\; \frac{\frac{1}{2}\log\left(Nm + N^2m^2\right) \,+ \log\left(\|\mu^0 - \mu^*\|_2\right) \,+ \log\left(\frac{1+d(E')}{\Delta(\theta;G)}\right)}{\log(1/\gamma)}.$$

This corollary follows by observing that the vector $(\mu^n - \mu^*)$ has less than $Nm + N^2m^2$ elements, so that $\|\mu^n - \mu^*\|_1 \leq \sqrt{Nm + N^2m^2}\,\|\mu^n - \mu^*\|_2$. Moreover, Theorems 3 and 4 provide an ℓ_1-ball radius such that the rounding schemes succeed (either with probability greater than $1 - \epsilon$, or deterministically) for all pseudomarginal vectors within these balls.

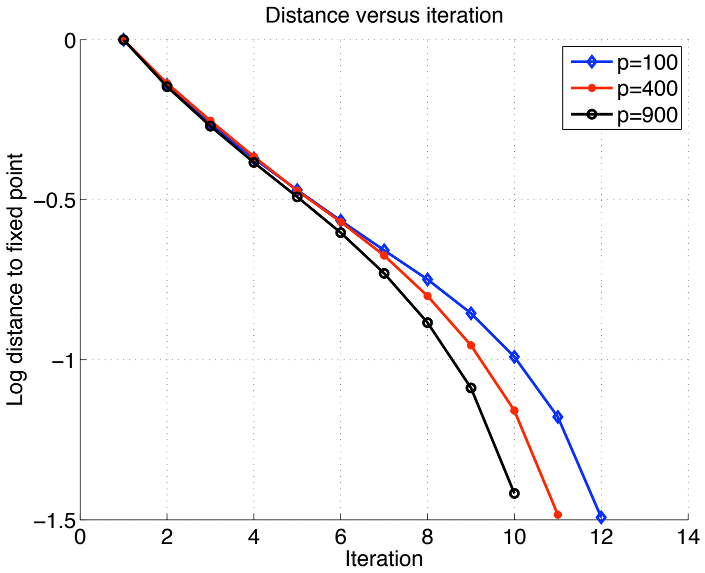

Figure 2. Plot of distance $\log_{10}\|\mu^n - \mu^*\|_2$ between the current entropic proximal iterate μ^n and the LP optimum μ^* versus iteration number for Potts models on grids with $p \in \{100, 400, 900\}$ vertices, $m = 5$ labels and SNR $= 1$. Note the superlinear rate of convergence.

5 Experiments

We performed experiments on a 4-nearest neighbor grid graphs with sizes varying from $N = 100$ to $p = 900$, using models with either $m = 3$ or $m = 5$ labels. The edge potentials were set to Potts functions, $\theta_{st}(x_s, x_t) = \beta_{st}\,\mathbb{I}[x_s = x_t]$, which penalize disagreement of labels by β_{st}. The Potts weights on edges β_{st} were chosen randomly as Uniform$(-1, +1)$, while the node potentials $\theta_s(x_s)$ were set as Uniform$(-\,\text{SNR}, \text{SNR})$, where the parameter SNR ≥ 0 controls the ratio of node to edge strengths, and thus corresponds roughly to a signal-to-noise ratio.

5.1 Rates of convergence

Figure 2 plots the logarithmic distance between the iterates μ^n of the entropic proximal method and the LP optimum μ^*, against the number of iterations for grids of different sizes. In all cases, note how the curves have an inverted quadratic shape, corresponding to superlinear convergence. Define the suboptimality factor of an iterate as the fraction of the energy of the iterate to the energy of the MAP optimum. Figure 3 plots the suboptimality factor of the entropic proximal iterates when rounded by the node-based randomized rounding scheme, against the number of iterations. Note again, the small number of iterations required for convergence.

5.2 Comparison of convergence rates

In Figure 4, we compare two of our proximal schemes—the entropic and the quadratic schemes—with a subgradient descent method (Feldman et al., 2002; Komodakis et al., 2007) and the max-product message passing algorithm. In particular, Komodakis et al. (2007) decompose the LP into a sum of tractable, e.g. tree-based, objective functions by duplicating the parameters. They then perform (sub)gradient ascent on the dual of this decomposed LP

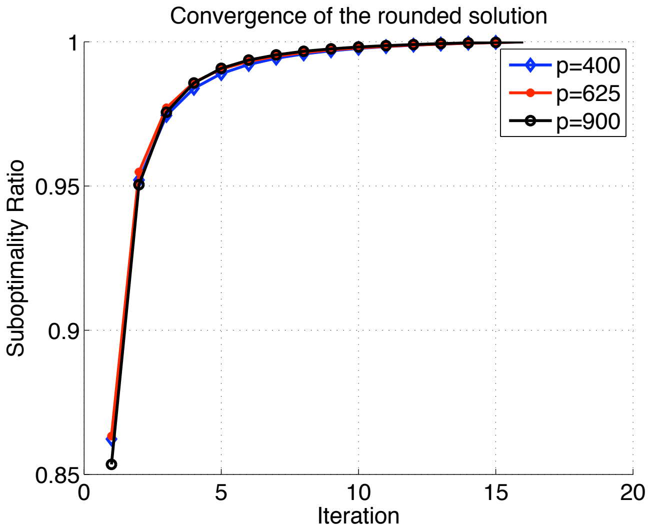

Figure 3. Plots of the fraction of the energy of the entropic proximal iterate μ^n when rounded by the node-based randomized rounding scheme to the energy of the MAP optimum μ^*; versus iteration number for Potts models on grids with $p \in \{100, 400, 900\}$ vertices, $m = 5$ labels and $\text{SNR} = 1$. Note the small number of iterations for convergence.

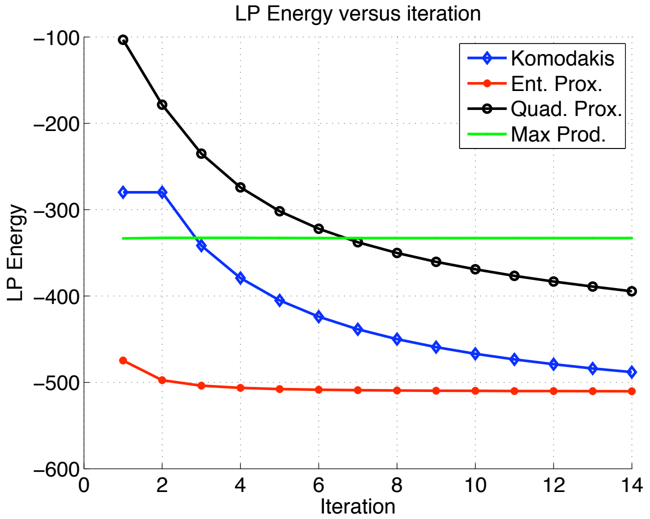

Figure 4. Plots comparing the convergence of the LP energy (that is, the negated LP objective) of the fractional solutions with the number of iterations for a Potts model with $N = 400$ vertices, $m = 3$ labels and $\text{SNR} = 2$. The methods of Komodakis et al. (2007), our entropic proximal method (Ent. Prox.), our quadratic proximal method (Quad. Prox.) and max product updates (Max. Prod.) are compared.

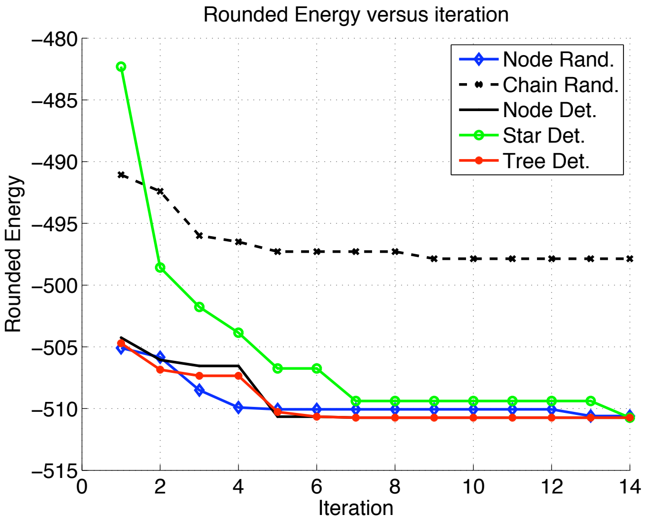

Figure 5. Plots comparing the convergence of the LP energy (that is, the negated LP objective) of the rounded entropic proximal solutions with the number of iterations. The node-based (Node Rand.) and chain-based (Chain Rand.) randomized rounding schemes, and the node-based (Node. Det.), neighborhood-based (Star Det.) and the tree-based (Tree Det.) deterministic rounding schemes are compared.

objective. For the comparison, we used a Potts model on a grid of 400 nodes, with each node taking 3 labels. The Potts weights were set as earlier, with SNR $= 2$. As Figure 4 shows, the entropic proximal scheme converges almost immediately, in six iterations, while the quadratic proximal converges quite a bit slower. The convergence rate of the subgradient ascent method lies between those of the entropic and the quadratic proximal schemes. As the figure shows, the max-product algorithm is stuck at a fixed point away from the LP optimum.

5.3 Comparison of rounding schemes

In Figure 5, we compare five of our rounding schemes on a Potts model on grid graphs with $N = 400$, $m = 3$ labels and SNR $= 2$. For the graph-structured randomized rounding schemes, we used the node-based rounding scheme (so that $E\backslash E' = \emptyset$), and the chain-based rounding scheme (so that $E\backslash E'$ is the set of horizontal chains in the grid). For the deterministic rounding schemes, we used the node-based, neighborhood-based and the tree-based rounding schemes. As the figure shows, the node-based randomized rounding scheme converges to an almost optimal solution almost immediately, in four iterations, closely followed by the node-based and tree-based deterministic rounding schemes.

6 Discussion

In this paper, we have developed distributed algorithms, based on the notion of proximal sequences, for solving graph-structured linear programming (LP) relaxations. Our methods respect the graph structure, and so can be scaled to large problems, and they exhibit a superlinear rate of convergence. We have also developed a series of graph-structured rounding schemes that can be used to generate integral solutions along with a certificate of optimality. These optimality certificates allow the algorithm to be terminated in a finite number of iterations.

The structure of our algorithms naturally lends itself to incorporating additional constraints, both linear and other types of conic constraints. It would be interesting to develop an adaptive version of our algorithm, which selectively incorporated new constraints as necessary, and then used the same proximal schemes to minimize the new conic program.

Acknowledgements

This work was partially supported by NSF grants CCF-0545862 and DMS-0528488.

A Detailed derivation of message-passing updates

In this appendix, we provided detailed derivation of the message-passing updates for the inner loops of the algorithms.

A.1 Derivation of Algorithm (2)

Consider the edge marginalization constraint for edge (s,t), $\mathbb{L}_i(G) \equiv \sum_{x_t} \mu_{st}(x_s, x_t) = \mu_s(x_s)$. Denoting the dual (Lagrange) parameter corresponding to the constraint by $\lambda_{st}(x_s)$, the Karush-Kuhn-Tucker conditions for the quadratic update (17) are given by

$$\begin{aligned}
\nabla q(\mu_{st}^{n,\tau+1}(x_s, x_t)) &= \nabla q(\mu_{st}^{n,\tau}(x_s, x_t)) + \lambda_{st}(x_s) \\
\nabla q(\mu_{s}^{n,\tau+1}(x_s)) &= \nabla q(\mu_{s}^{n,\tau}(x_s)) - \lambda_{st}(x_s) \\
\mu_{st}^{n,\tau+1}(x_s, x_t) &= \mu_{st}^{n,\tau}(x_s, x_t) + \lambda_{st}(x_s) \\
\mu_{s}^{n,\tau+1}(x_s) &= \mu_{s}^{n,\tau}(x_s) - \lambda_{st}(x_s),
\end{aligned}$$

while the constraint itself gives

$$\sum_{x_t} \mu_{st}^{n,\tau+1}(x_s, x_t) = \mu_s^{n,\tau}(x_s) \tag{43}$$

Solving for $\lambda_{st}(x_s)$ yields equation (19a). The node marginalization follows similarly, so that overall, we obtain message-passing Algorithm (2) for the inner loop.

A.2 Derivation of Algorithm (3)

The projection $\mu^{n,\tau+1} = \Pi_h(\mu^{n,\tau}, \mathbb{L}_i(G))$ onto the individual constraint $\mathbb{L}_i(G)$ is defined by the optimization problem:

$$\mu^{n,\tau+1} \;=\; \min_{\mathbb{L}_i(G)} \{h(\mu) - \mu^\top \nabla h(\mu^{n,\tau})\}.$$

Consider the subset $\mathbb{L}_i(G)$ defined by the marginalization constraint along edge (s,t), namely $\sum_{x'_t \in \mathcal{X}} \mu_{st}(x_s, x'_t) = \mu_s(x_s)$ for each $x_s \in \mathcal{X}$. Denoting the dual (Lagrange) parameters corresponding to these constraint by $\lambda_{st}(x_s)$, the KKT conditions are given by

$$\nabla h(\mu_{st}^{n,\tau+1}(x_s, x_t)) = \nabla h(\mu_{st}^{n,\tau}(x_s, x_t)) + \lambda_{st}(x_s), \quad \text{and} \tag{44a}$$

$$\nabla h(\mu_{s}^{n,\tau+1}(x_s)) = \nabla h(\mu_{s}^{n,\tau}(x_s)) - \lambda_{st}(x_s). \tag{44b}$$

Computing the gradient ∇h and performing some algebra yields the relations

$$\begin{aligned}
\mu_{st}^{(n,\tau+1)}(x_s, x_t) &= \mu_{st}^{(n,\tau)}(x_s, x_t) \exp(\lambda_{st}^{(n,\tau+1)}(x_s)), && (45a)\\
\mu_s^{(n,\tau+1)}(x_s) &= \mu_s^{(n,\tau)}(x_s) \exp(-\lambda_{st}^{(n,\tau+1)}(x_s)), \quad \text{and} && (45b)\\
\exp(2\lambda_{st}^{(n,\tau+1)}(x_s)) &= \frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_t} \mu_{st}^{(n,\tau)}(x_s, x_t)}, && (45c)
\end{aligned}$$

from which the updates (22) follow.

Similarly, for the constraint set defined by the node marginalization constraint $\sum_{x_s \in \mathcal{X}} \mu_s(x_s) = 1$, we have $\nabla h(\mu_s^{(n,\tau+1)}(x_s)) \ = \ \nabla h(\mu_s^{(n,\tau)}(x_s)) + \lambda_s^{(n,\tau+1)}$, from which

$$\begin{aligned}
\mu_s^{(n,\tau+1)}(x_s) &= \mu_s^{(n,\tau)}(x_s) \exp(\lambda_s^{(n,\tau+1)}), \qquad \text{and} && (46a)\\
\exp(\lambda_s^{(n,\tau+1)}) &= 1/\sum_{x_s \in \mathcal{X}} \mu_s^{(n,\tau)}(x_s). && (46b)
\end{aligned}$$

The updates in equation (24) follow.

B Proof of Lemma 1

We provide a detailed proof for the entropic scheme; the arguments for other proximal algorithms are analogous. The key point is the following: regardless of how the proximal updates are computed, they must necessary the necessary Lagrangian conditions for optimal points over the set $\mathbb{L}(G)$. Accordingly, we define the following sets of Lagrange multipliers:

λ_{ss}	for the normalization constraint $C_{ss}(\mu_s) = \sum_{x'_s} \mu_s(x'_s) - 1 = 0$
$\lambda_{st}(x_s)$	for the marginalization constraint $C_{ts}(x_s) = \sum_{x'_t} \mu_{st}(x_s, x'_t) - \mu_s(x_s) = 0$
$\gamma_{st}(x_s, x_t)$	for the non-negativity constraint $\mu_{st}(x_s, x_t) \geq 0$.

(There is no need to enforce the non-negativity constraint $\mu_s(x_s) \geq 0$ directly, since it is implied by the non-negativity of the joint pseudo-marginals and the marginalization constraints.)

With this notation, consider the Lagrangian associated with the entropic proximal update at step n:

$$L(x; \lambda, \gamma) \quad = \quad C(\mu; \theta, \mu^n) + \langle \gamma, \, \mu \rangle + \sum_{s \in V} \lambda_{ss} C_{ss}(x_s) + \sum_{(s,t) \in E} \big[\lambda_{ts}(x_s) C_{ts}(x_s) + \lambda_{st}(x_t) C_{st}(x_t)\big],$$

where $C(\mu; \theta, \mu^n)$ is shorthand for the cost component $-\langle \theta, \, \mu \rangle + \frac{1}{\omega^n} D_\alpha(\mu \,\|\, \mu^n)$. Using C, C' to denote constants (whose value can change from line to line), we now take derivatives to find the necessary Lagrangian conditions:

$$\begin{aligned}
\frac{\partial L}{\partial \mu_s(x_s)} &= -\theta_s(x_s) + \frac{2\alpha_s}{\omega^n} \log \frac{\mu_s(x_s)}{\mu_s^n(x_s)} + C + \lambda_{ss} + \sum_{t \in N(s)} \lambda_{ts}(x_s), \qquad \text{and}\\
\frac{\partial L}{\partial \mu_{st}(x_s, x_t)} &= -\theta_{st}(x_s, x_t) + \frac{2\alpha_{st}}{\omega^n} \log \frac{\mu_{st}(x_s, x_t)}{\mu_{st}^n(x_s, x_t)} + C' + \gamma_{st}(x_s, x_t) - \lambda_{ts}(x_s) - \lambda_{st}(x_t).
\end{aligned}$$

Solving for the optimum $\mu = \mu^{n+1}$ yields

$$\begin{aligned}
\frac{2\alpha_s}{\omega^n} \log \mu_s^{n+1}(x_s) &= \theta_s(x_s) + \frac{2\alpha_s}{\omega^n} \log \mu_s^n(x_s) - \sum_{t \in N(s)} \lambda_{ts}(x_s) + C \\
\frac{2\alpha_{st}}{\omega^n} \log \mu_{st}^{n+1}(x_s, x_t) &= \theta_{st}(x_s, x_t) + \frac{2\alpha_{st}}{\omega^n} \log \mu_{st}^n(x_s, x_t) - \gamma_{st}(x_s, x_t) \\
&\quad + \lambda_{ts}(x_s) + \lambda_{st}(x_t) + C'.
\end{aligned}$$

From these conditions, we can compute the energy invariant (36):

$$\begin{aligned}
\frac{2}{\omega^n} F(x; \mu^{n+1}) &= \sum_{s \in V} \frac{2\alpha_s}{\omega^n} \log \mu_s^{n+1}(x_s) + \sum_{(s,t) \in E} \frac{2\alpha_{st}}{\omega^n} \log \mu_{st}^{n+1}(x_s, x_t) + C \\
&= F(x; \theta) + \frac{2}{\omega^n} \Big\{ \sum_{s \in V} \alpha_s \log \mu^n(x_s) + \sum_{(s,t) \in E} \alpha_{st} \log \mu_{st}^n(x_s, x_t) \Big\} \\
&\quad - \sum_{(s,t) \in E} \gamma_{st}(x_s, x_t) + C \\
&= F(x; \theta) + \frac{2}{\omega^n} F(x; \mu^n) - \sum_{(s,t) \in E} \gamma_{st}(x_s, x_t) + C.
\end{aligned}$$

Now since $\mu^n > 0$, by complementary slackness, we must have $\gamma_{st}(x_s, x_t) = 0$, which implies that

$$\frac{2}{\omega^n} F(x; \mu^{n+1}) = F(x; \theta) + \frac{2}{\omega^n} F(x; \mu^n) + C. \tag{47}$$

From this equation, it is a simple induction to show for some constants $\gamma_n > 0$ and $C_n \in \mathbb{R}$, we have $F(x; \mu^n) = \gamma_n F(x; \theta) + C_n$ for all iterations $n = 1, 2, 3, \ldots$, which implies preservation of the maximizers. If at iteration $n = 0$, we initialize $\mu^0 = 0$ to the all-uniform distribution, then we have $\frac{2}{\omega^1} F(x; \mu^1) = F(x; \theta) + C'$, so the statement follows for $n = 1$. Suppose that it holds at step n; then $\frac{2}{\omega^n} F(x; \mu^n) = \frac{2}{\omega^n} \gamma_n F(x; \theta) + \frac{2C_n}{\omega_n}$, and hence from the induction step (47), we have $F(x; \mu^{n+1}) = \gamma_{n+1} F(x; \theta) + C_{n+1}$, where $\gamma_{n+1} = \frac{\omega^n}{2} \gamma_n$.

C Proof of Theorem 2

Consider the expected cost of the configuration $X(\mu^n; E')$ obtained from the randomized rounding procedure of Algorithm 5. A simple computation shows that

$$\mathbb{E}[F(X(\mu^n; E'); \theta)] = G(\bar{\mu}) := \sum_{i=1}^{K} H(\mu^n; T_i) + H(\mu^n; E')$$

where

$$H(\mu^n; T_i) := \sum_{s \in V_i} \sum_{x_s} \mu_s^n(x_s) \theta_s(x_s) + \sum_{(s,t) \in E_i} \sum_{x_s, x_t} \mu_{st}^n(x_s, x_t) \theta_{st}(x_s, x_t), \tag{48a}$$

$$H(\mu^n; E') := \sum_{(u,v) \in E'} \sum_{x_s, x_t} \mu_u^n(x_u) \mu_v^n(x_v) \theta_{st}(x_u, x_v). \tag{48b}$$

We now show by induction that the derandomized rounding scheme achieves cost at least as large as this expected value. Let $\bar{\mu}^{(i)}$ denote the updated pseudomarginals at the end of the i-th iteration. Since we initialize with $\bar{\mu}^{(0)} = \mu^n$, we have $G(\bar{\mu}^{(0)}) = \mathbb{E}[F(X(\mu^n; E'); \theta)]$. Consider the i-th step of the algorithm; the algorithm computes the portion of the derandomized solution $x^d_{V_i}$ over the $i-$th tree. It will be convenient to use the decomposition $G = G_i + G_{\backslash i}$, where

$$G_i(\bar{\mu}) := \sum_{s \in V_i} \sum_{x_s} \bar{\mu}_s(x_s) \Big\{ \theta_s(x_s) + \sum_{\{t \mid (s,t) \in E'\}} \sum_{x_t} \bar{\mu}_t(x_t) \theta_{st}(x_s, x_t) \Big\} + \sum_{(s,t) \in E_i} \sum_{x_s, x_t} \bar{\mu}_{st}(x_s, x_t)\, \theta_{st}(x_s, x_t),$$

and $G_{\backslash i} = G - G_i$. If we define

$$\overline{F}_i(x_{V_i}) := \sum_{s \in V_i} \big\{ \theta_s(x_s) + \sum_{t \colon (s,t) \in E'} \sum_{x_t} \bar{\mu}_t^{(i-1)}(x_t) \theta_{st}(x_s, x_t) \big\} + \sum_{(s,t) \in E_i} \theta_{st}(x_s, x_t),$$

it can be seen that $G_i(\bar{\mu}^{(i-1)}) = \mathbb{E}[\overline{F}_i(x_{V_i})]$ where the expectation is under the tree-structured distribution over X_{V_i} given by

$$p(x_{V_i}; \bar{\mu}^{(i-1)}(T_i)) \;=\; \prod_{s \in V_i} \bar{\mu}^{(i-1)}(x_s) \prod_{(s,t) \in E_i} \frac{\bar{\mu}^{(i-1)}(x_s, x_t)}{\bar{\mu}^{(i-1)}(x_s) \bar{\mu}^{(i-1)}(x_t)}.$$

Thus when the algorithm makes the choice $x^d_{V_i} = \arg\max_{x_{V_i}} \overline{F}_i(x_{V_i})$, it holds that

$$G_i(\bar{\mu}^{(i-1)}) \;=\; \mathbb{E}[\overline{F}_i(x_{V_i})] \;\leq\; \overline{F}_i(x^d_{V_i}).$$

The updated pseudomarginals $\bar{\mu}^{(i)}$ at the end the i-th step of the algorithm are given by,

$$\bar{\mu}^{(i)}_s(x_s) \;=\; \begin{cases} \bar{\mu}^{(i-1)}_s(x_s) & \text{if } s \notin V_i \\ 0 & \text{if } s \in V_i, X_{d,s} \neq x_s \\ 1 & \text{if } s \in V_i, X_{d,s} = x_s \end{cases}$$

$$\bar{\mu}^{(i)}_{st}(x_s, x_t) \;=\; \begin{cases} \bar{\mu}^{(i-1)}_{st}(x_s, x_t) & \text{if } (s,t) \notin E_i \\ \bar{\mu}^{(i)}_s(x_s) \bar{\mu}^{(i)}_t(x_t) & \text{if } (s,t) \in E_i \end{cases}$$

In other words, $\bar{\mu}^{(i)}(T_i)$ is the indicator vector of the maximum energy subconfiguration $x^d_{V_i}$. Consequently, we have

$$G_i(\bar{\mu}^{(i)}) \;=\; \overline{F}_i(x^d_{V_i}) \;\geq\; G_i(\bar{\mu}^{(i-1)}),$$

and $G_{\backslash i}(\bar{\mu}^{(i)}) = G_{\backslash i}(\bar{\mu}^{(i-1)})$, so that at the end of the i-th step, $G(\bar{\mu}^{(i)}) \geq G(\bar{\mu}^{(i-1)})$. By induction, we conclude that $G(\bar{\mu}^{(K)}) \geq G(\bar{\mu}^{(0)})$, where K is the total number of trees in the rounding scheme.

At the end of K steps, the quantity $\bar{\mu}^{(K)}$ is the indicator vector for $x^d(\mu^n; E')$ so that $G(\bar{\mu}^{(K)}) = F(X_d(\mu^n; E'); \theta)$. We have also shown that $G(\bar{\mu}^{(0)}) = \mathbb{E}[F(X(\mu^n; E'); \theta)]$. Combining these pieces, we conclude that $F(x^d(\mu^n; E'); \theta) \geq \mathbb{E}[F(X(\mu^n; E'); \theta)]$, thereby completing the proof.

D Proof of Theorem 3

Let $p_{\text{succ}} = \mathbb{P}[X(\mu^n; E') = x^*]$, and let $R(\mu^n; E')$ denote the (random) integral vertex of $\mathbb{L}(G)$ that is specified by the random integral solution $X(\mu^n; E')$. (Since E' is some fixed forest-inducing subset, we frequently shorten this notation to $R(\mu^n)$.) We begin by computing the expected cost of the random solution, where the expectation is taken over the rounding procedure. A simple computation shows that $\mathbb{E}[\langle \theta,\, R(\mu^n)\rangle] \;:=\; \sum_{i=1}^{K} H(\mu^n; T_i) + H(\mu^n; E')$, where $H(\mu^n; T_i)$ and $H(\mu^n; E')$ were defined previously (48).

We now upper bound the difference $\langle \theta,\, \mu^*\rangle - \mathbb{E}[\langle \theta,\, R(\mu^n)\rangle]$. For each subtree $i = 1, \ldots, K$, the quantity $D_i := H(\mu^*; T_i) - H(\mu^n; T_i)$ is upper bounded as

$$\begin{aligned} D_i &= \sum_{s \in V_i} \sum_{x_s} \Big[\mu^*_s(x_s) - \mu^n_s(x_s)\Big] \theta_s(x_s) + \sum_{(s,t) \in E_i} \sum_{x_s, x_t} \Big[\mu^*_s(x_s)\mu^*_t(x_t) - \mu^n_{st}(x_s, x_t)\Big] \theta_{st}(x_s, x_t) \\ &\leq \sum_{s \in V_i} \delta_s(\theta) \sum_{x_s} |\mu^*_s(x_s) - \mu^n_s(x_s)| + \sum_{(s,t) \in E_i} \delta_{st}(\theta) \sum_{x_s, x_t} |\mu^*_{st}(x_s, x_t) - \mu^n(x_s, x_t)|. \end{aligned}$$

In asserting this inequality, we have used the fact that that the matrix with entries given by $\mu^*_s(x_s)\mu^*_t(x_t) - \mu^n_{st}(x_s, x_t)$ is a difference of probability distributions, meaning that all its entries are between -1 and 1, and their sum is zero.

Similarly, we can upper bound the difference $D(E') = H(\mu^*; E') - H(\mu^n; E')$ associated with E':

$$\begin{aligned} D(E') &= \sum_{(u,v) \in E'} \sum_{x_u, x_v} \Big[\mu^*_u(x_u)\mu^*_v(x_v) - \mu^n_u(x_u)\mu^n_v(x_v)\Big] \theta_{uv}(x_u, x_v) \\ &\leq \sum_{(u,v) \in E'} \delta_{uv}(\theta) \sum_{x_u, x_v} \Big|\mu^*_u(x_u)\mu^*_v(x_v) - \mu^n_u(x_u)\mu^n_v(x_v)\Big| \\ &\leq \sum_{(u,v) \in E'} \delta_{uv}(\theta) \sum_{x_u, x_v} \Big\{ \Big|\mu^*_u(x_u)[\mu^*_v(x_v) - \mu^n_v(x_v)]\Big| + \Big|\mu^n_v(x_v)[\mu^*_u(x_u) - \mu^n_u(x_u)]\Big| \Big\} \\ &\leq \sum_{(u,v) \in E'} \delta_{uv}(\theta) \Big\{ \sum_{x_u} |\mu^n_u(x_u) - \mu^*_u(x_u)| + \sum_{x_u} |\mu^n_v(x_v) - \mu^*_v(x_v)| \Big\}. \end{aligned}$$

Combining the pieces, we obtain

$$\begin{aligned} \langle \theta,\, \mu^*\rangle - \mathbb{E}[\langle \theta,\, R(\mu^n)\rangle] &\leq \delta_G(\theta) \Big\{ \|\mu^n - \mu^*\|_1 + \sum_{s \in V} d(s; E') \sum_{x_s} |\mu^n_s(x_s) - \mu^*_s(x_s)| \Big\} \\ &\leq (1 + d(E'))\delta_G(\theta) \|\mu^n - \mu^*\|_1. \end{aligned} \tag{49}$$

In the other direction, we note that when the rounding fails, then we have

$$\langle \theta,\, \mu^*\rangle - \langle \theta,\, R(\mu^n)\rangle \;\geq\; \max_{x \neq x^*} [F(x^*; \theta) - F(x; \theta)].$$

Consequently, conditioning on whether the rounding succeeds or fails, we have

$$\begin{aligned} \langle \theta,\, \mu^*\rangle - \mathbb{E}[\langle \theta,\, R(\mu^n)\rangle] &\geq p_{\text{succ}} \left[\langle \theta,\, \mu^*\rangle - \langle \theta,\, \mu^*\rangle\right] + (1 - p_{\text{succ}}) \max_{x \neq x^*} [F(x^*; \theta) - F(x; \theta)] \\ &= (1 - p_{\text{succ}}) \max_{x \neq x^*} [F(x^*; \theta) - F(x; \theta)]. \end{aligned}$$

Combining this lower bound with the upper bound (49), performing some algebra, and using the definition of the gap $\Delta(\theta; G)$ yields that the probability of successful rounding is at least

$$p_{\text{succ}} \quad \geq \quad 1 - \frac{(1 + d(E'))}{\Delta(\theta; G)} \|\mu^n - \mu^*\|_1.$$

If the condition (41) holds, then this probability is at least $1 - \epsilon$, as claimed.

E Proof of Theorem 4

The proof follows that of Theorem 3 until equation (49), which gives

$$\langle \theta,\, \mu^* \rangle - \mathbb{E}[\langle \theta,\, R(\mu^n) \rangle] \quad \leq (1 + d(E'))\, \delta_G(\theta)\, \|\mu^n - \mu^*\|_1.$$

Let $v^d(\mu^n; E')$ denote the integral vertex of $\mathbb{L}(G)$ that is specified by the derandomized integral solution $x^d(\mu^n; E')$. Since E' is some fixed forest-inducing subset, we frequently shorten this notation to $v^d(\mu^n)$. Theorem 2 shows that

$$\mathbb{E}[\langle \theta,\, R(\mu^n) \rangle] \leq \langle \theta,\, v^d(\mu^n) \rangle.$$

Suppose the derandomized solution is not optimal so that $v^d(\mu^n) \neq \mu^*$. Then, from the definition of the graph-based gap $\Delta(\theta; G)$, we obtain

$$\langle \theta,\, \mu^* \rangle - \langle \theta,\, v^d(\mu^n) \rangle \geq \delta_G(\theta)\, \Delta(\theta; G)$$

Combining the pieces, we obtain

$$\begin{aligned} \delta_G(\theta)\, \Delta(\theta; G) \quad &\leq \quad \langle \theta,\, \mu^* \rangle - \langle \theta,\, v^d(\mu^n) \rangle \\ &\leq \quad \langle \theta,\, \mu^* \rangle - \mathbb{E}[\langle \theta,\, R(\mu^n) \rangle] \\ &\leq \quad (1 + d(E'))\delta_G(\theta)\|\mu^n - \mu^*\|_1, \end{aligned}$$

which implies $\|\mu^n - \mu^*\|_1 \;\geq\; \frac{\Delta(\theta;G)}{1+d(E')}$. However, this conclusion is a contradiction under the given assumption on $\|\mu^n - \mu^*\|_1$ in the theorem. It thus holds that the derandomized solution $v^d(\mu^n)$ is equal to the MAP optimum μ^*, thereby completing the proof.

References

M. Bayati, D. Shah, and M. Sharma. Maximum weight matching for max-product belief propagation. In *Int. Symp. Info. Theory*, Adelaide, Australia, September 2005.

M. Bayati, C. Borgs, J. Chayes, and R. Zecchina. Belief-propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions. Technical Report arxiv:0709.1190, Microsoft Research, September 2007.

U. Bertele and F. Brioschi. *Nonserial dynamic programming.* Academic Press, New York, 1972.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, Boston, MA, 1997.

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

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

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

S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, UK, 2004.

Y. Boykov, O. Veksler, and R .Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

Y. Censor and S. A. Zenios. *Parallel Optimization - Theory, Algorithms and Applications.* Oxford University Press, 1997.

C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625, 2005.

F. Deutsch and H. Hundal. The rate of convergence for the cyclic projection algorithm I: Angles between convex sets. *Journal of Approximation Theory*, 142:36 55, 2006.

J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo-like codes and its relation to iterative approaches. In *Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing*, October 2002.

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.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6:721–741, 1984.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Neural Information Processing Systems*, Vancouver, Canada, December 2007.

D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51, 1989.

T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energy. In *The 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, Helsinki, 2008.

B. Huang and T. Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In *AISTATS*, San Juan, Puerto Rico, March 2007.

A. N. Iusem and M. Teboulle. Convergence rate analysis of nonquadratic proximal methods for convex and linear programming. *Mathematics of Operations Research*, 20(3):657–677, 1995.

J. K. Johnson, D. M. Malioutov, and A. S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Proc. 45th Allerton Conf. Comm. Cont. Comp.*, Urbana-Champaign, IL, September 2007.

J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *IEEE Symp. Found. Comp. Science*, pages 14–24, 1999.

V. Kolmogorov. Convergent tree-reweighted message-passing for energy minimization. In *International Workshop on Artificial Intelligence and Statistics*, January 2005.

V. Kolmogorov and M. J. Wainwright. On optimality properties of tree-reweighted message-passing. In *Uncertainty in Artificial Intelligence*, July 2005.

N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, Rio di Janeiro, Brazil, 2007.

V. K. Koval and M. I. Schlesinger. Two-dimensional programming in image analysis problems. *USSR Academy of Science, Automatics and Telemechanics*, 8:149–168, 1976. In Russian.

P. Kumar, P.H.S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming. *IEEE Conference of Computer Vision and Pattern Recognition*, 2006.

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

P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 737–744, 2006.

S. Sanghavi, D. Shah, and A. Willsky. Message-passing for max-weight independent set. In *Neural Information Processing Systems*, Vancouver, Canada, December 2007.

M.V. Solodov and B.F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22:1013–1035, 2001.

P. Tseng and D. P. Bertsekas. On the convergence of the exponential multiplier method for convex programming. *Math. Programming*, 60:1—19, 1993.

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

M. J. Wainwright and M. I. Jordan. Treewidth-based conditions for exactness of the Sherali-Adams and Lasserre relaxations. Technical report, UC Berkeley, Department of Statistics, No. 671, September 2004.

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.

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the max-product algorithm and its generalizations. *Statistics and Computing*, 14:143–166, April 2004.

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing. *IEEE Trans. Information Theory*, 51(11):3697–3717, November 2005.

Y. Weiss, C. Yanover, and T. Meltzer. Map estimation, linear programming and belief propagation with convex free energies. In *Uncertainty in Artificial Intelligence*, 2007.

C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation: An empirical study. *Journal of Machine Learning Research*, 7:1887–1907, September 2006.

J.S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. Info. Theory*, 51(7):2282–2312, July 2005.