Message-passing for Graph-structured Linear Programs: Proximal Projections, Convergence 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

Abstract

A large body of past work has focused on the first-order tree-based LP relaxation for the MAP problem in Markov random fields. This paper develops a family of super-linearly convergent LP solvers based on proximal minimization schemes using Bregman divergences that exploit the underlying graphical structure, and so scale well to large problems. All of 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. The inner loop updates are distributed and respect the graph structure, and thus can be cast as message-passing algorithms. We establish various convergence guarantees for our algorithms, illustrate their performance, and also present rounding schemes with provable optimality guarantees.

1. Introduction

A key computational challenge associated with discrete Markov random fields (MRFs) is the problem of *maximum a posteriori* (MAP) estimation: computing the most probable configuration(s). For general graphs, this MAP problem includes a large number of classical NP-complete problems, including MAX-CUT independent set, and satisfiability problems, among various others.

This intractability motivates the development and analysis of methods for obtaining approximate solutions. The ordinary max-product algorithm is a form of non-serial dynamic-programming, exact for trees, and also widely used as a heuristic for obtaining approximate solutions to the MAP problem, but it suffers from convergence failures, and despite some local optimality results (Freeman & Weiss, 2001), it has no general correctness guarantees. For certain MRFs arising in computer vision, Boykov et al. (2001) studied graph-cut based search algorithms that compute a local maximum over two classes of moves. A related class of methods are those based on various types of convex relaxations, in which the discrete MAP problem is relaxed some type of convex optimization problem over continuous variables. Examples include linear programming (LP) relaxations (Wainwright et al., 2005; Chekuri et al., 2005), as well as quadratic, semidefinite and other conic programming relaxations (Ravikumar & Lafferty, 2006; Kumar et al., 2006; Wainwright & Jordan, 2003).

Among convex relaxations, LP relaxation is the least computationally expensive and best understood. The primary focus of this paper is a well-known tree-based LP relaxation (Chekuri et al., 2005; Wainwright et al., 2005) of the MAP estimation problem for pairwise Markov random fields, based on optimizing over a set of locally consistent pseudomarginals on edges and vertices of the graph. In principle, this LP relaxation can be solved by any standard solver, including simplex or interior-point methods (Bertsimas & Tsitsiklis, 1997). However, such generic methods fail to exploit the graph-structured nature of the LP, and hence do not scale favorably to large-scale problems.

Wainwright et al. (2005) established a connection between this tree-based LP relaxation and the class of tree-reweighted max-product (TRW-MP) algorithms, showing that suitable TRW-MP fixed points specify optimal solutions to the LP relaxation. Subsequent work has extended this basic connection in various in-

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

teresting ways. For instance, Kolmogorov (2005) developed a serial form of TRW-MP with some convergence properties but as with the ordinary TRW-MP updates, no guarantees of LP optimality. Weiss et al. (2007) connected convex forms of sum-product and exactness of reweighted max-product algorithms. Globerson and Jaakkola (2007) developed a convergent dual-ascent algorithm, but its fixed points are guaranteed to be LP-optimal only for binary problems, as is also the case for the TRW-MP algorithm (Kolmogorov & Wainwright, 2005), and the rate of convergence is not analyzed. Other authors (Komodakis et al., 2007; Feldman et al., 2002) have proposed subgradient methods, but such methods typically have sub-linear convergence rates.

The goal of this paper is to develop and analyze various classes of message-passing algorithms that always solve the LP, and are provably convergent with at least a geometric rate. The methods that we develop are flexible, in that new constraints can be incorporated in a relatively seamless manner, with new messages introduced to enforce them. All of the algorithms in this paper are based on the notion 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 reweighted Bethe entropies. For all choices, we show how the intermediate proximal problems can be solved by message-passing updates on the graph, guaranteed to converge but with a distributed nature that scales favorably. An additional desirable feature, given the wide variety of lifting methods for further constraining LP relaxations (Wainwright & Jordan, 2003), is that additional constraints are easily incorporated within the framework.

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 $\mathcal{X} = \{0, 1, 2, \ldots, m\}$, let $X = \{X_1, \ldots, X_p\} \in \mathcal{X}^p$ denote a $p-$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 $\theta_s : \mathcal{X} \to$ and $\theta_{st} : \mathcal{X} \times \mathcal{X} \to$ be singleton and edgewise potential functions respectively, we assume that the distribution takes the form $\mathbb{P}(x; \theta) \propto \exp\big\{\sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)\big\}$.

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

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

This problem is an integer program, since it involves optimizing over the discrete space $\mathcal{X}^p$. The functions $\theta_s(\cdot)$ and $\theta_{st}(\cdot)$ can always be represented in the form

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

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

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 basic linear programming (LP) relaxation 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:

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

$$\sum_{x_t} \mu_{st}(x_s, x_t) = \mu_s(x_s) \qquad \text{for all } (s,t) \in E.$$

The polytope defined in this way is denoted LOCAL(G), or $\mathbb{L}(G)$ for short. The LP relaxation is based on solving maximizing the linear function

$$\sum_s \sum_{x_s} \theta_s(x_s)\mu_s(x_s) + \sum_{(s,t) \in E} \sum_{x_s, x_t} \theta_{st}(x_s, x_t)\mu_{st}(x_s, x_t),$$

subject to the constraint $\mu \in \mathbb{L}(G)$. In the sequel, we write this LP more compactly in the form $\max_{\mu \in \mathbb{L}(G)} \theta^T \mu$. 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.

3. Proximal Minimization Schemes

We begin by defining the notion of a proximal minimization scheme, and the Bregman divergences that we use to define our proximal sequences. Instead of referring to the maximization problem $\max_{\mu \in \mathbb{L}(G)} \theta^T \mu$, it is convenient to consider the equivalent minimization problems $\min_{\mu \in \mathbb{L}(G)} -\theta^T \mu$.

3.1. Proximal Minimization

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

$$\mu^{n+1} = \arg \min_{\mu \in \mathbb{L}(G)} \left\{ -\theta^T \mu + \frac{1}{\omega^n} D_f(\mu \,\|\, \mu^n) \right\}, \qquad (4)$$

where for each $n = 0, 1, 2, \ldots$, μ^n denotes current iterate, $\{\omega^n\}$ denotes a sequence of positive weights, and D_f is a certain type of generalized distance, known as the proximal function. The purpose of introducing the proximal function is to convert the original LP—a convex optimization problem but non-differentiable in dual space —into a strictly convex optimization problem that can be solved relatively easily. This scheme appears similar to an annealing scheme, in that it involves a choice of weights $\{\omega^n\}$. However, although the weights $\{\omega^n\}$ can be adjusted for faster convergence, they can also be set to a constant, unlike for annealing procedures, which would typically require that $1/\omega^n \to 0$. The reason is that $D_f(\mu \,\|\, \mu^{(n)})$, as a generalized distance, itself converges to zero when the method gets closer to 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 & Tsitsiklis, 1997; Iusem & Teboulle, 1995).

In this paper, we focus exclusively on proximal functions that are Bregman divergences (Censor & Zenios, 1997), a class that includes various well-known divergences (e.g., quadratic norm, Kullback-Leibler divergence etc.). More specifically, we say that a function f is a Bregman function if it is continuously differentiable, strictly convex, and has bounded level sets. It then induces a Bregman divergence

$$D_f(\mu \,\|\, \nu) := f(\mu) - f(\nu) - \langle \nabla f(\nu), \mu - \nu \rangle \qquad (5)$$

This function satisfies $D_f(\mu \,\|\, \nu) \geq 0$ with equality iff $\mu = \nu$, but need not be symmetric or satisfy the triangle inequality, so it is known as a generalized distance.

We study the sequence $\{\mu^n\}$ of proximal iterates (4) for the following choices of Bregman divergences.

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

$$Q(\mu \,\|\, \nu) := \sum_{s \in V} \|\mu_s - \nu_s\|^2 + \sum_{(s,t) \in E} \|\mu_{st} - \nu_{st}\|^2, \qquad (6)$$

where we have used the shorthand

$$\|\mu_s - \nu_s\|^2 = \sum_{x_s} |\mu_s(x_s) - \nu_s(x_s)|^2,$$

and similarly for the edges. The Bregman function this corresponds to is the quadratic function,

$$f(\mu) = \frac{1}{2} \left\{ \sum_{s, x_s} \mu_s^2(x_s) + \sum_{s,t,x_s,x_t} \mu_{st}^2(x_s, x_t) \right\} \qquad (7)$$

Weighted Entropic Distances: Here we consider a (possibly weighted) sum of Kullback-Leibler (KL) divergences across the nodes and edges:

$$D(\mu \,\|\, \nu) = \sum_{s \in V} \rho_s D(\mu_s \,\|\, \nu_s) + \sum_{s,t} \rho_{st} D(\mu_{st} \,\|\, \nu_{st}) \quad (8)$$

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, and $\{\rho_s, \rho_{st}\}$ are positive node and edge weights, respectively. An advantage of the KL distance, in contrast to the quadratic norm, is that it automatically acts to enforce non-negativity constraints on the pseudomarginals. The Bregman function this corresponds to is the entropy function,

$$f(\mu) = \sum_s H_s(\mu_s) + \sum_{s,t} H_{st}(\mu_{st}) \qquad (9)$$

where H_s and H_{st} are singleton and edge-based entropies, respectively.

An extension of define a Bregman function based on a convex combination of tree-structured entropy functions (Wainwright & Jordan, 2003), and using expressions such as the reweighted Bethe entropy which are equivalent to the convex combination of tree entropies within the local polytope, we can derive an iterative procedure involving tree-reweighted message passing to solve the outer proximal steps. We defer further details to a full-length version.

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 the sequence of each proximal minimization 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 based on a decomposition of the constraint set LOCAL(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

$$\widehat{\mu} \quad := \quad \arg\min_{\mu \in C} D_f(\mu \,\|\, \nu) \tag{10}$$

By taking derivatives and using standard conditions for optima over convex sets (Bertsekas & Tsitsiklis, 1997), the defining optimality condition for $\widehat{\mu}$ is

$$\left(\nabla f(\widehat{\mu}) - \nabla f(\nu)\right)^T \left(\mu - \widehat{\mu}\right) \quad \geq \quad 0 \tag{11}$$

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

$$\min_{\mu \in \mathbb{L}(G)} \left\{ -\theta^T \mu + \frac{1}{\omega^n} D_f(\mu \,\|\, \mu^n) \right\}. \tag{12}$$

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

$$\left(\nabla f(\mu^{n+1}) - \nabla f(\mu^n) - \omega^n \theta\right)^T \left(\mu - \mu^{n+1}\right) \quad \geq \quad 0$$

for all $\mu \in C$. Note that these optimality conditions are of the same form as the Bregman projection conditions (11), 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 (11) can be leveraged to compute the proximal update (12). 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 & Zenios, 1997).

To simplify notation, for any Bregman function f, let us define the operator

$$\mathrm{J}_f(\mu, \nu) = (\nabla f)^{-1}(\nabla f(\mu) + \nu)$$

and for any Bregman divergence D with Bregman function f and any convex set C, define the projection operator

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

With this notation, we can write the proximal update in a compact manner as a type of projection

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

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 & 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 procedure 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 $\mu_s^{(0)}(x_s) = \frac{1}{m}$, $\mu_{st}^{(0)}(x_s, x_t) = \frac{1}{m^2}$.
• Outer loop: For iterations $n = 0, 1, 2, \ldots$, update $\mu^{n+1} = \Pi_D\left(\mathrm{J}_f(\mu^n, \omega^n \theta); \mathbb{L}(G)\right)$.
    – Solve outer loop via Inner loop:
        (a) Initialize $\mu^{n,0} = \mathrm{J}_f(\mu^n, \omega^n \theta)$.
        (b) For $\tau = 0, 1, 2, \ldots$, set $i(\tau) = \tau \mod T$.
        (c) Set $\mu^{n,\tau+1} = \Pi_D\left(\mu^{n,\tau}; \mathbb{L}_{i(\tau)}(G)\right)$.
```

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 & Tsitsiklis, 1997), combined with convergence guarantees for cyclic Bregman projections (Censor & Zenios, 1997). In the following section, we derive the message-passing updates corresponding to various Bregman functions of interest. We also give rates of convergence for the cyclic projection updates in the inner loop.

3.3. Quadratic Projections

Consider the proximal sequence with the quadratic distance Q from equation (6); the Bregman function inducing this distance is the quadratic function $f(y) = \frac{1}{2}y^2$, whose gradient is given by $\nabla f(y) = y$.

The Map $\nu = \mathbf{J}_f(\mu, \omega\theta)$: In this case, it can be derived as,

$$\nabla f(\nu) = \nabla f(\mu) + \omega\theta \tag{13}$$

$$\Rightarrow \nu = \mu + \omega\theta \tag{14}$$

whence we get the initialization in Equation 18.

The Projections $\mu^{n,\tau+1} = \Pi_Q(\mu^{n,\tau}, \mathbb{L}_i(G))$: onto the individual constraints $\mathbb{L}_i(G)$; the associated local update takes the form

$$\mu^{n,\tau+1} = \min_{\alpha \in \mathbb{L}_i(G)} \{f(\alpha) - \alpha^\top \nabla f(\mu^{n,\tau})\} \tag{15}$$

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 KKT conditions for (15) are given by

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

while the constraint itself gives

$$\sum_{x_t} \mu_{st}^{n,\tau+1}(x_s,x_t) = \mu_s^{n,\tau}(x_s) \tag{17}$$

Solving for $\lambda_{st}(x_s)$ yields equation (20). The node marginalization follows similarly, so that overall, we obtain message-passing algorithm (2) for the inner loop.

3.4. Entropic Projections

Consider the proximal sequence with the Kullback-Leibler distance $D(\mu \,\|\, \nu)$ defined in equation (8); the Bregman function inducing the distance is a sum of negative entropy functions $f(\mu) = \mu \log \mu$, and its gradient is given by $\nabla f(\mu) = \log(\mu) + \mathbf{1}$.

The derivation of the updates mirrors the previous section, and defering the details to a full-length version, we get the message passing algorithm (3) for the inner loop.

There are also interesting similarities between our corresponding dual updates and sum-product updates—which are updates to the dual parameters—details of which we defer to a full-length version of this paper due to lack of space.

3.5. Reweighted Entropy Projections

The message passing updates here are “reweighted” versions of those in the previous section for the unweighted entropy induced Kullback-Leibler divergence proximal iterates.

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

```
Initialization:
  $\mu_{st}^{(n,0)}(x_s,x_t) = \mu_{st}^{(n)}(x_s,x_t) + w^n\theta_{st}(x_s,x_t)$   (18)
  $\mu_s^{(n,0)}(x_s) = \mu_s^{(n)}(x_s) + w^n\theta_s(x_s)$   (19)
repeat
  for each edge $(s,t) \in E$ do
    $\mu_{st}^{(n,\tau+1)}(x_s,x_t) = \mu_{st}^{(n,\tau)}(x_s,x_t) + (1/L+1)\left(\mu_s^{(n,\tau)}(x_s) - \sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)\right)$   (20)
    $\mu_s^{(n,\tau+1)}(x_s) = \mu_s^{(n,\tau)}(x_s) + (1/L+1)\left(-\mu_s^{(n,\tau)}(x_s) + \sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)\right)$   (21)
  end for
  for each node $s \in V$ do
    $\mu_s^{(k+1)}(x_s) = \mu_s^{(k)}(x_s) + \frac{1}{L}(1 - \sum_{x_s}\mu_s^{(k)}(x_s))$
    $\mu_s^{(k+1)}(x_s) = \max(0, \mu_s^{(k+1)}(x_s))$
  end for
until convergence
```

Initialization of Proximal Steps:

$$\mu_{st}^{(n,0)}(x_s,x_t) = \mu_{st}^{(n)}(x_s,x_t)\exp(\omega^n/\rho_{st}\theta_{st}(x_s,x_t))$$

$$\mu_s^{(n,0)}(x_s) = \mu_s^{(n)}(x_s)\exp(\omega^n/\rho_s\theta_s(x_s)).$$

Projections: The node normalization update remains the same as in the previous section, while the marginalization update changes as,

$$\mu_{st}^{(n,\tau+1)}(x_s,x_t) = \mu_{st}^{(n,\tau)}(x_s,x_t)\left(\frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)}\right)^{\frac{\rho_s}{\rho_s+\rho_{st}}}$$

$$\mu_s^{(n,\tau+1)}(x_s) = \mu_s^{(n,\tau)}(x_s)^{\frac{\rho_s}{\rho_s+\rho_{st}}}\left(\sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)\right)^{\frac{\rho_{st}}{\rho_s+\rho_{st}}}$$

4. Rounding with Optimality Certificates

A key practical issue in applying LP relaxation is how round the fractional solution; a standard approach is

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

```
Initialization:
  $\mu_{st}^{(n,0)}(x_s,x_t) = \mu_{st}^{(n)}(x_s,x_t)\exp(\omega^n\theta_{st}(x_s,x_t))$
  $\mu_s^{(n,0)}(x_s) = \mu_s^{(n)}(x_s)\exp(\omega^n\theta_s(x_s))$
repeat
  for each edge $(s,t) \in E$ do
    $\mu_{st}^{(n,\tau+1)}(x_s,x_t) = \mu_{st}^{(n,\tau)}(x_s,x_t)\sqrt{\frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)}}$
    $\mu_s^{(n,\tau+1)}(x_s) = \sqrt{\mu_s^{(n,\tau)}(x_s)\sum_{x_t}\mu_{st}^{(n,\tau)}(x_s,x_t)}$
  end for
  for each node $s \in V$ do
    $\mu_s^{(n,\tau+1)}(x_s) = \frac{\mu_s^{(n,\tau)}(x_s)}{\sum_{x_s}\mu_s^{(n,\tau)}(x_s)}$
  end for
until convergence
```

to round the node marginals to the nearest integer solution. However, in general, such rounding procedures need not always output the optimal integer configuration. An attractive feature of our proximal Bregman procedures is the existence of rounding schemes which, assuming that the LP relaxation is tight, can produce the LP integral optimum and certify that it is correct, even before the pseudomarginals converge to the LP solution. Here we describe two rounding schemes, and state the optimality certificate associated with each.

Node-based Rounding: This method applies to any of our proximal schemes. Given the vector μ^n of pseudomarginals at iteration n, define an integer configuration x^n by choosing, for each vertex $s \in V$, a value $x_s^n \in \arg\max_{x_s}\mu_s^n(x_s)$. Say that such a rounding is *edgewise-consistent* if for all edges $(s,t) \in E$, we have $(x_s^n, x_t^n) \in \arg\max_{(x_s,x_t)}\mu_{st}^n(x_s,x_t)$.

Tree-based Rounding: We describe this method in application to the unweighted entropic proximal updates. Let $T_1,\ldots,T_M$ be a set of spanning trees that cover the graph (meaning that each edge appears in at least one tree); for each edge (s,t), define the edge weight $\alpha_{st} = \frac{1}{M}\sum_{i=1}^M \mathbb{I}[(s,t) \in T_i]$. Then for each tree $i = 1,\ldots,M$:

(a) Define the tree-structured energy function $E_i(x) := \sum_s \mu^n(x_s) + \sum_{(s,t)\in E(T_i)}\frac{1}{\alpha_{st}}\mu_{st}^n(x_s,x_t)$.

(b) Run the ordinary max-product problem on energy $E_i(x)$ 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.

The following result characterizes the optimality guarantees associated with these rounding schemes:

Theorem 1 (Rounding with optimality certificates)**.** *At any iteration $n = 1, 2, \ldots$, any edge-consistent configuration obtained from node-rounding, or any tree-consistent configuration obtained from tree-rounding is guaranteed to be MAP optimal for the original problem.*

The proof is based on a certain energy-invariance property of the proximal updates; in particular, at any iteration n, the pseudomarginals μ^n have an associated function $F(x;\mu^n)$ which is proportional to the energy $E(\theta;x) = \sum_s \theta_s(x_s) + \sum_{st}\theta_{st}(x_s,x_t)$ of the graphical model. For instance, for the entropic proximal scheme, at any iteration n, the function $F(x;\mu^n) := \prod_{s\in V}\mu_s^n(x_s)\ \prod_{(s,t)\in E}\mu_{st}^n(x_s,x_t)$ is proportional to the exponential of $E(\theta;x)$. (See the technical report (Ravikumar et al., 2008) for full details.)

Both rounding schemes require relatively little computation. Of course, the node-rounding scheme is purely local, and so trivial to implement. With reference to the tree-rounding scheme, many graphs can be covered with a small number M of trees (e.g. $M = 2$ for grid graphs). Consequently, the tree-rounding scheme requires running the ordinary max-product algorithm twice, certainly more expensive than node-rounding but doable. In practice, we find that tree-rounding tends to find LP optima more quickly than node rounding.

5. Convergence Rates

The convergence of our message passing updates follows from two sets of results: (a) convergence of proximal algorithms (Bertsekas & Tsitsiklis, 1997) and (b) convergence of cyclic Bregman projections (Censor & Zenios, 1997). Our outer loop is a proximal algorithm; which has been well-studied in the optimization literature. A sequence $\mu^{(t)}$ is said to have superlinear convergence to the optimum μ^* if $\lim_{k\to\infty}\frac{\|\mu^{(t+1)}-\mu^*\|}{\|\mu^{(t)}-\mu^*\|} = 0$. Note that such convergence is faster than a multiplicative contraction ($\lim_{k\to\infty}\frac{\|\mu^{(t+1)}-\mu^*\|}{\|\mu^{(t)}-\mu^*\|} \leq \alpha < 1$). Bertsekas and Tsistiklis (1997) show that a proximal algorithm with a quadratic proximity has a superlinear convergence under mild conditions, whereas Iusem and Teboulle (1995) show the same for the entropy proximity. Under the assumption that inner loops are solved exactly, these convergence results then show that our outer iterates converge superlinearly. Our inner loop message updates use cyclic Bregman projections; Censor and Zenios (1997) show that with dual feasibility correction, projections onto general convex sets are convergent. For Euclidean (quadratic) projections onto linear constraints (half-spaces), 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 are, the faster the convergence; for instance, a single iteration suffices for completely orthogonal constraints. Our inner updates thus converge geometrically to an $\epsilon-$suboptimal solution for any outer proximal step. As noted earlier, the proximal convergence results assume that the inner loop has been solved exactly, while the Bregman projection results yield geometric convergence to but an $\epsilon-$suboptimal solution. While with ϵ small enough, e.g. 10^{-6} as in our experiments, this issue might not be practically relevant, there has been some recent work, e.g. (Solodov & Svaiter, 2001), showing that under mild conditions, superlinear rates still hold for $\epsilon-$suboptimal proximal iterates.

6. Experiments

We performed experiments on a 4-nearest neighbor grid graphs with sizes varying from $p = 100$ to $p = 900$, in all cases using models with 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$(-\,\mathrm{SNR}, \mathrm{SNR})$, where the parameter $\mathrm{SNR} \geq 0$ controls the ratio of node to edge strengths, and thus corresponds roughly to a signal-to-noise ratio.

Figure 1 shows plots of the logarithmic distance between the current iterate μ^n and the LP optimum μ^n for grids of different sizes. In all cases, note how the curves have an inverted quadratic shape, corresponding to superlinear convergence.

Figure 2 shows the fraction of edges for which the node-based rounding is edgewise inconsistent for grids of different sizes. Note how the fraction converges to zero in a small number of iterations. Figure 3 shows the fraction of the energy of the rounded solution to the energy of the MAP optimum, or the suboptimality factor. Note again, the small number of iterations for convergence.

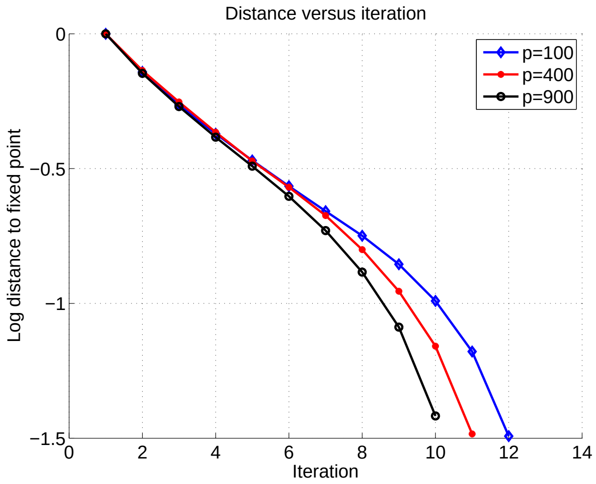

Figure 1. Plot of distance $\log_{10} \|\mu^n - \mu^*\|_2$ between the current iterate μ^n and the LP optimum μ^* versus iteration number for Potts models on grids with $p \in \{100, 400, 900\}$ vertices, and $\mathrm{SNR} = 1$. Note the superlinear rate of convergence.

7. 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 also developed 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 Work supported by NSF grants CCF-0545862 and DMS-0528488. We thank the anonymous reviewers for helpful comments.

References

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

Bertsimas, D., & Tsitsikilis, J. (1997). *Introduction to linear optimization*. Belmont, MA: Athena Scientific.

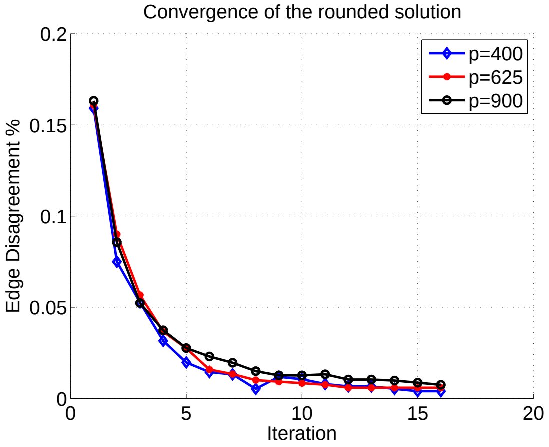

Figure 2. Plots of the fraction of edges for which the node-based rounding is edgewise inconsistent for grids of different sizes. Recall that when the fraction is zero, we recover the MAP optimum.

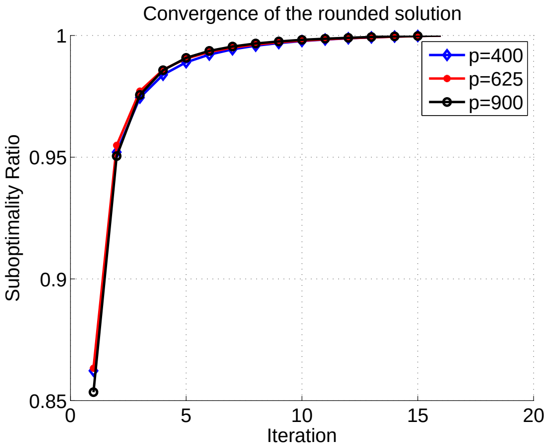

Figure 3. Plots of the fraction of the energy of the rounded solution to the energy of the MAP optimum. Note the small number of iterations for convergence.

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

Censor, Y., & Zenios, S. A. (1997). *Parallel optimization - theory, algorithms and applications*. Oxford University Press.

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

Deutsch, F., & Hundal, H. (2006). The rate of convergence for the cyclic projections algorithm i: Angles between convex sets. *Journal of Approximation Theory*, *142*, 36 55.

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

Freeman, W. T., & Weiss, Y. (2001). On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. Info. Theory*, *47*, 736–744.

Globerson, A., & Jaakkola, T. (2007). Fixing max-product: Convergent message passing algorithms for map lp-relaxations. *Neural Information Processing Systems* (p. To appear). Vancouver, Canada.

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

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

Kolmogorov, V., & Wainwright, M. J. (2005). On optimality properties of tree-reweighted message-passing. *UAI*.

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

Kumar, P., Torr, P., & Zisserman, A. (2006). Solving markov random fields using second order cone programming. *IEEE CVPR*.

Ravikumar, P., Agarwal, A., & Wainwright, M. J. (2008). *Message-passing for graph-structured linear programs: Proximal projections, convergence and rounding schemes* (Technical Report). UC Berkeley.

Ravikumar, P., & Lafferty, J. (2006). Quadratic programming relaxations for metric labeling and markov random field map estimation. *ICML '06* (pp. 737–744).

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

Wainwright, M., Jaakkola, T., & Willsky, A. (2005). Map estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, *51*, 3697–3717.

Wainwright, M. J., & Jordan, M. I. (2003). *Graphical models, exponential families, and variational inference* (Technical Report). UC Berkeley, Department of Statistics, No. 649.

Weiss, Y., Yanover, C., & Meltzer, T. (2007). Map estimation, linear programming and belief propagation with convex free energies. *UAI*.