Tree Block Coordinate Descent for MAP in Graphical Models

David Sontag **Tommi Jaakkola**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

A number of linear programming relaxations have been proposed for finding most likely settings of the variables (MAP) in large probabilistic models. The relaxations are often succinctly expressed in the dual and reduce to different types of reparameterizations of the original model. The dual objectives are typically solved by performing local block coordinate descent steps. In this work, we show how to perform block coordinate descent on spanning trees of the graphical model. We also show how all of the earlier dual algorithms are related to each other, giving transformations from one type of reparameterization to another while maintaining monotonicity relative to a common objective function. Finally, we quantify when the MAP solution can and cannot be decoded directly from the dual LP relaxation.

1 Introduction

Many important practical problems can be solved with graphical models, such as clustering, molecular conformations, stereopsis, or haplotype inference. One of the inference problems in these models is finding the most likely setting of the variables (the MAP assignment) given observed data. The complexity of finding a MAP assignment in general depends on the tree-width of the graph. Unfortunately, in many real problems such as stereopsis, the graph tree-width lies beyond the tractable range.

A common way to approximate the MAP problem is through the use of Linear Programming (LP) relaxations. In this case, the MAP problem is first cast as an integer programming problem, and subsequently relaxed to a linear program by removing the integer constraints. Whenever the relaxed solution is integral (corresponds to an assignment), it is guaranteed to be the optimal solution. In special cases (e.g., ferromagnetic models and matching problems), the MAP assignment can be found efficiently using LP relaxations, even for large tree-width graphs.

In recent years, a number of dual LP relaxation algorithms have been proposed, and these have been demonstrated to be useful tools for solving large MAP problems (Kolmogorov, 2006; Werner, 2007; Globerson & Jaakkola, 2008; Komodakis & Paragios, 2008). These algorithms can be understood as dual coordinate descent steps, but operate in different duals of the same pairwise LP relaxation. The dual coordinate descent algorithms are designed from the point of view of introducing local distributed operations. This is indeed often advantageous for large problems. However, such operations may take a long time to converge even if the model is exactly solvable, such as a tree. In this paper, we show how the dual coordinate descent operations can be carried out for a tree in one block step. The resulting block update takes linear time in the size of the tree, and closely resembles max-product.

We place these dual algorithms under a common framework so that they can be understood as optimizing the same objective, and demonstrate how to change from one representation to another in a monotone fashion relative to the common objective. This framework permits us to analyze and extend all the methods together as a group. For example, the block update can be used in combination with any of the specific dual algorithms through the transformations we introduce. One of the key goals in introducing the new framework is to facilitate the design of new algorithms and modifications that can be used broadly across the different dual formulations. We exemplify this by pro-

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

viding a monotone version of the TRW algorithm that makes use of the new tree-block update. Moreover, we discuss parallel yet monotone update schemes for the distributed coordinate descent steps.

Finally, the algorithms are only as good as their ability to reconstruct a MAP assignment (primal solution) from the dual optimal solution. We provide conditions for when the MAP assignment can and cannot be found from the dual solutions. The analysis applies to all of the dual algorithms.

2 MAP and its LP Relaxation

We consider the MAP problem for pairwise MRFs. The model is given by a graph $G = (V, E)$ with vertices V and edges E. Each edge $ij \in E$ is associated with a potential function $\theta_{ij}(x_i, x_j)$. The goal is to find an assignment $\mathbf{x} = \{x_i\}_{i \in V}$ that maximizes

$$\theta(\mathbf{x}) = \sum_{ij \in E} \theta_{ij}(x_i, x_j). \tag{1}$$

For notational simplicity, we assume that single node potentials are absorbed into the edge potentials.

A popular approximate solution to the MAP problem is obtained by turning the integer programming problem into a simpler LP problem through a *pairwise relaxation*. For each edge and assignment to the variables on the edge we have the marginal $\mu_{ij}(x_i, x_j) \geq 0$, such that $\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) = 1$. The pairwise relaxation is given by

$$\max_{\mathbf{x}} \theta(\mathbf{x}) \leq \max_{\mu \in M_L} \left\{ \sum_{ij \in E} \sum_{x_i, x_j} \theta_{ij}(x_i, x_j) \mu_{ij}(x_i, x_j) \right\} \tag{2}$$

where the *local marginal polytope* M_L enforces that edge marginals are consistent with each other, i.e.

$$\sum_{x_i} \mu_{ij}(x_i, x_j) \quad = \quad \sum_{x_k} \mu_{jk}(x_j, x_k) \tag{3}$$

for all pairs of edges ij and jk with a common node j. If the solution to the pairwise LP relaxation is integral then it is a MAP solution, i.e., the inequality in Eq. (2) is tight. Otherwise, the objective upper bounds the value of the MAP. For binary MRFs, even when only part of the solution is integral, a MAP solution is guaranteed to exist that extends the partial integral assignment.

3 Dual LPs

In this section we introduce a common framework for understanding several dual linear programs corresponding to LP relaxations. All of the dual LPs that we will discuss in Section 5 can be viewed as minimizing the pairwise functional

$$J(f) = \sum_i \max_{x_i} f_i(x_i) + \sum_{ij \in E} \max_{x_i, x_j} f_{ij}(x_i, x_j) \tag{4}$$

over possible decompositions of the original function to single node $f_i(x_i)$ and pairwise $f_{ij}(x_i, x_j)$ potentials. The necessary constraint on these potentials is that they define *sup-reparameterizations*:

$$F(\theta) = \left\{ f : \begin{array}{c} \forall \mathbf{x}, \sum_i f_i(x_i) + \sum_{ij \in E} f_{ij}(x_i, x_j) \\ \geq \sum_{ij \in E} \theta_{ij}(x_i, x_j) \end{array} \right\} \tag{5}$$

Without any other constraints on $F(\theta)$, the optimum of this LP would give the MAP value, i.e.

$$\max_{\mathbf{x}} \theta(\mathbf{x}) = \min_{f \in F(\theta)} J(f). \tag{6}$$

For example, one optimal solution $f^* \in F(\theta)$ that attains the MAP value is obtained from $f^*_{ij}(x_i, x_j) = \max_{\mathbf{x} \setminus \{x_i, x_j\}} \left\{ \sum_{ij \in E} \theta_{ij}(x_i, x_j) \right\} / |E|$ and $f^*_i(x_i) = 0$. The dual $\min_{f \in F(\theta)} J(f)$, used by Komodakis & Paragios (2008), has one constraint for every assignment ensuring that the reparameterization's value on $\mathbf{x}$ is at least as large as $\theta(\mathbf{x})$. Not surprisingly, finding the optimum of this LP is NP-hard.

The key to understanding the different LP formulations are the *additional constraints* that are imposed on $F(\theta)$. It can be shown that simply changing the inequality constraint in Eq. (5) to an equality constraint would result in this being the dual of the pairwise relaxation. In the remainder of this section, we will specify three different but related constrained classes, each of which corresponds to a (different) dual of the pairwise LP relaxation.

The first class is a simple reparameterization in terms of single node potentials: $F_L(\theta)$ is defined as

$$\left\{ f : \begin{array}{l} f_i(x_i) = \sum_{j \in N(i)} \delta_{ji}(x_i), \\ f_{ij}(x_i, x_j) = \theta_{ij}(x_i, x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j), \\ \text{for some } \{\delta_{ji}(x_i)\} \end{array} \right\} \tag{7}$$

The single node "messages" $\delta_{ji}(x_i)$ are subtracted from the edge terms and added to the node terms so as to maintain a valid reparameterization: $\sum_i f_i(x_i) + \sum_{ij \in E} f_{ij}(x_i, x_j) = \sum_{ij \in E} \theta_{ij}(x_i, x_j)$ for all $\mathbf{x}$.

It is straightforward to show that $\min_{f \in F_L(\theta)} J(f)$ is the dual of the pairwise LP relaxation given by Eq. (2) and (3). $F_L(\theta)$ is the same as the dual linear program introduced by Schlesinger *et al.* in 1976 and optimized by the *max-sum diffusion algorithm* (see Werner, 2007, and references within).

We also introduce a restricted version of $F_L(\theta)$ where the single node potentials are identically zero: $f_i(x_i) =$

1. Obtain max-marginals μ by running max-product on $f_T = \{f_i(x_i), f_{ij}(x_i, x_j) : ij \in T\}$.
2. Update the parameters for $ij \in T$ as follows:

$$f_i^{(t+1)}(x_i) = \frac{1}{n} \log \mu_i(x_i)$$
$$f_{ij}^{(t+1)}(x_i, x_j) = \log \mu_{ij}(x_i, x_j) - \frac{n_{j\to i}}{n} \log \mu_i(x_i) - \frac{n_{i\to j}}{n} \log \mu_j(x_j).$$

Figure 1: **Max-Product Tree Block Update Alg.** n is the number of nodes in tree, and $n_{j\to i}$ is the number of nodes in the subtree of node i with parent j.

$\sum_{j\in N(i)} \delta_{ji}(x_i) = 0$. This corresponds to an additional constraint on how $\delta_{ji}(x_i)$ can be chosen, i.e., they must sum to zero around each node. We call this set of single node (zero) and pairwise potentials $F_{L,E}(\theta)$ as the objective only depends on the edges. Clearly, $F_{L,E}(\theta) \subseteq F_L(\theta)$. An algorithm similar to max-sum diffusion can be given to optimize this representation.

Finally, we introduce a complementary class where the edge terms are zero and the $f_i(x_i)$ are defined in terms of constrained "messages" $\delta_{ji}(x_i)$ as follows:

$$F_{L,V}(\theta) = \left\{ f : \begin{array}{l} f_i(x_i) = \sum_{j\in N(i)} \delta_{ji}(x_i) \\ f_{ij}(x_i, x_j) = 0 \\ \delta_{ji}(x_i) + \delta_{ij}(x_j) \geq \theta_{ij}(x_i, x_j) \end{array} \right\} \quad (8)$$

It is easy to see that $F_{L,V}(\theta) \subseteq F(\theta)$. However, in general, potentials in $F_{L,V}(\theta)$ are not members of $F_L(\theta)$. Minimizing $J(f)$ subject to $f \in F_{L,V}(\theta)$ is the dual formulation given by Komodakis & Paragios (2008), and obtains the same value as the pairwise relaxation. It is also closely related to the dual given by Globerson & Jaakkola (2008). We will make this precise in Section 5.

4 Tree Block Updates

Most coordinate-descent algorithms for solving the dual LPs perform local operations on the messages, updating edge reparameterizations or messages around each node. This is advantageous for simplicity. However, even when the model is a tree, a large number of local operations may be needed to reach a dual optimal solution. In this section, we provide block update algorithms for trees, analogous to exact collect-distribute computation on a tree, but leading to a reparameterization rather than max-marginals.

In each step of the algorithm we isolate a tree out of the current reparameterization objective and perform a block update for this tree. Such a tree block update can lead to faster convergence for appropriately chosen trees, and may also help in decoding, as discussed in Section 6.

We give two tree block update algorithms. The first algorithm is shown in Figure 1. Consider any tree structured model specified by node parameters $f_i(x_i), i \in T$, and edge parameters $f_{ij}(x_i, x_j)$, $ij \in T$. This tree may have been extracted from the current LP dual to perform a block update. The algorithm works by running a forward and backward pass of max-product to compute the max-marginals $\mu_{ij}(x_i, x_j) = \max_{\mathbf{x}\setminus\{i,j\}} \Pr(\mathbf{x})$ for the tree distribution

$$\Pr(\mathbf{x}) = \frac{1}{Z(f_T)} \exp\left\{ \sum_{i\in T} f_i(x_i) + \sum_{ij\in T} f_{ij}(x_i, x_j) \right\},$$

and then uses the max-marginals to construct a reparameterization of the original tree model. After each update, the constant $c = \log(Z(f_T^{(t)})/Z(f_T^{(t+1)}))$ should be added to the dual objective. This can be found by evaluating $f_T^{(t)}(\mathbf{x}) - f_T^{(t+1)}(\mathbf{x})$ for any assignment $\mathbf{x}$.

This approach uses only the standard max-product algorithm to solve the LP relaxation. If applied, for example, with stars around individual nodes rather than spanning trees, this results in one of the simplest dual-coordinate descent algorithms given to date. However, since the tree block updates are used as part of the overall dual LP algorithm, it is important to ensure that the effect of the updates is distributed to subsequent operations as effectively as possible. We found that by instead performing tree block updates directly within the class of $F_L(\theta)$ reparameterizations, we obtain significantly faster running times.

The second block update algorithm, shown in Figure 2, finds a set of $\delta_{ji}(x_i)$ messages such that $f_T^{(t+1)} \in F_L(f_T)$, defined by the $\delta_{ji}(x_i)$, minimizes $J(f_T)$. This algorithm makes use of *directed* trees. We found that choosing a random root works well. The first step is to send max-product messages from the leaves to the root. Then, on the downward pass, we do a series of edge reparameterizations, constructing the $\delta_{ji}(x_i)$ to ensure that each term in the objective is maximized by the MAP assignment. ($c = 0$ for this algorithm.)

Proposition 4.1. *The tree block procedures given in Figures 1 and 2 attain the MAP value for a tree,*

$$\max_{\mathbf{x}} \sum_{i\in T} f_i^{(t)}(x_i) + \sum_{ij\in T} f_{ij}^{(t)}(x_i, x_j) = J(f_T^{(t+1)}) + c,$$

Proof. (Sketch). **Max-product algorithm.** First we show this returns a reparameterization. Using the fact

1. Select a root, and orient the edges away from it.
2. Evaluate max-marginal messages toward the root (collect operation). For each $j \in ch(i)$,
$$m_{j\to i}(x_i) = \max_{x_j}\Big[\ f_{ij}(x_i,x_j) + f_j(x_j) + \sum_{k\in ch(j)} m_{k\to j}(x_j)\Big].$$
3. Reparameterize away from the root (distribute operation). Set $\tilde{n}_i = n_i$. For each $j \in ch(i)$:
   (a) Define $m^{-j}(x_i) = \sum_{k\in N(i)\setminus j} m_{k\to i}(x_i)$.
   (b) Set
$$\delta_{ji}(x_i) = -\frac{n_j}{\tilde{n}_i}\left[m^{-j}(x_i)+f_i(x_i)\right] + \frac{\tilde{n}_i - n_j}{\tilde{n}_i}\max_{x_j}\Big[f_{ij}(x_i,x_j)+f_j(x_j)+m^{-i}(x_j)\Big],$$
$$\delta_{ij}(x_j) = \frac{n_j - \tilde{n}_i}{\tilde{n}_i}\left[m^{-i}(x_j)+f_j(x_i)\right] + \frac{n_j}{\tilde{n}_i}\max_{x_i}\Big[f_{ij}(x_i,x_j)+f_i(x_i)+m^{-j}(x_i)\Big].$$
   (c) Re-define upward $m_{j\to i}(x_i) = \delta_{ji}(x_i)$. Define downward $m_{i\to j}(x_j) = \delta_{ij}(x_j)$. Change size of subtree: $\tilde{n}_i = \tilde{n}_i - n_j$.
4. Update the parameters for $ij \in T$ as follows:
$$f_i^{(t+1)}(x_i) = f_i(x_i) + \sum_{j\in N(i)} \delta_{ji}(x_i)$$
$$f_{ij}^{(t+1)}(x_i,x_j) = f_{ij}(x_i,x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j).$$

Figure 2: **Sequential Tree Block Update Alg.** $ch(i)$ denotes the children of i in the tree, and $N(i)$ all neighbors of i in the tree. Also, n_i is the number of nodes in the subtree rooted at i.

that $(\sum_{j\in N(i)} n_{j\to i} - 1)/n = |N(i)| - 1$, we get that $\exp\{\sum_{i\in T} f_i^{(t+1)}(x_i) + \sum_{ij\in T} f_{ij}^{(t+1)}(x_i,x_j)\}$ equals

$$\prod_{ij\in T}\mu_{ij}(x_i,x_j)\prod_{i\in T}\mu_i(x_i)^{1-|N(i)|} \tag{9}$$

which, by a special case of the junction-tree theorem, can be shown to be proportional to $\Pr(\mathbf{x})$.

Let $\mathbf{x}^*$ be the maximizing assignment of $f_T^{(t)}(\mathbf{x})$. Clearly x_i^* is a maximizer of $f_i^{(t+1)}(x_i)$. We now show that x_i^*, x_j^* also maximizes $f_{ij}^{(t+1)}(x_i,x_j)$. Note that $f_{ij}^{(t+1)}(x_i^*,x_j^*) = 0$ since $\mu_{ij}(x_i^*,x_j^*) = \mu_i(x_i^*)$ and $n_{j\to i} + n_{i\to j} = n$. However, $\mu_{ij}(x_i,x_j) \le \mu_i(x_i)$ for any x_i, x_j, which implies that $f_{ij}^{(t+1)}(x_i,x_j) \le 0$.

Sequential algorithm. For a tree, the pairwise LP relaxation is tight, so the dual optimal reparameterization necessarily attains the MAP value. Let $\delta_i = \{\delta_{ji}(x_i)\}_{j\in N(i)}$ denote the reparameterization around node i. We can write the objective as a function of the messages as $J(f_T^{(t+1)}) = J(\delta_1,\ldots,\delta_n)$. Set node 1 as the root. The collect operation in the algorithm corresponds to evaluating $\min_{\delta_2,\ldots,\delta_n} J(\delta_1,\ldots,\delta_n)$, which can be done recursively and corresponds to evaluating max-marginal messages. Note that the LP relaxation is also tight for each subtree. δ_1 is then solved exactly at the root node via a series of edge reparameterizations. In the distribute phase of the algorithm, we iteratively instantiate each reparameterization by minimizing $\min_{\delta_{i+1},\ldots,\delta_n} J(\hat{\delta}_1,\ldots,\hat{\delta}_{i-1},\delta_i,\delta_{i+1},\ldots,\delta_n)$ with respect to δ_i when $\hat{\delta}_1,\ldots,\hat{\delta}_{i-1}$ are already fixed. □

4.1 A Monotone Version of TRW

The tree-reweighted max-product algorithm (TRW) (Wainwright *et al.*, 2005) for MAP problems iteratively combines inference operations carried out on trees of the graph to solve the pairwise LP relaxation. In this section, we show that by using the tree block updates, the TRW algorithm becomes monotone in the dual LP.

Consider a collection of trees T of G, and a distribution over these trees given by $\rho(T)$. For example, we could give equal weight to a few trees that together cover all of the edges in the graph. The dual variables are parameter vectors $\theta^T(\mathbf{x}) = \sum_{i\in T}\theta_i^T(x_i) + \sum_{ij\in T}\theta_{ij}^T(x_i,x_j)$ for each tree T. The TRW dual problem is to minimize $\sum_T \rho(T)\max_{\mathbf{x}}\theta^T(\mathbf{x})$ subject to the reparameterization constraint $\sum_T \rho(T)\theta^T(\mathbf{x}) = \theta(\mathbf{x})$.

The tree-based update algorithm for TRW given by Wainwright *et al.* (2005) is shown below. ρ_{ij} denotes the probability that an edge $ij \in E$ is part of a tree drawn from the distribution $\rho(T)$.

1. For each tree T, set $\theta_i^T(x_i) = \theta_i(x_i)/\rho_i$ and $\theta_{ij}^T(x_i,x_j) = \theta_{ij}(x_i,x_j)/\rho_{ij}$.
2. Reparameterize each tree distribution. TRW does this by computing max-marginals μ for $\Pr(\mathbf{x};\theta^T)$ using max-product, then setting
$$\hat{\theta}_i^T(x_i) = \log\mu_i(x_i) \tag{10}$$
$$\hat{\theta}_{ij}^T(x_i,x_j) = \log\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)}$$
3. Average the solutions, and return to Step 1:
$$\hat{\theta}_i(x_i) = \sum_{T:i\in T}\rho(T)\hat{\theta}_i^T(x_i) \tag{11}$$
$$\hat{\theta}_{ij}(x_i,x_j) = \sum_{T:ij\in T}\rho(T)\hat{\theta}_{ij}^T(x_i,x_j).$$

Kolmogorov (2006) showed that TRW does not monotonically solve the dual LP. However, if in Step 2 we replace max-product with either of our tree block update algorithms (Figure 1 or 2) applied to θ^T, we obtain a monotone algorithm. With the max-product tree block update, the new algorithm looks nearly identical to TRW. The following proposition shows that these modified TRW steps are indeed valid and monotone with respect to the common objective $J(\theta)$.

Proposition 4.2. *Steps 1-3 described above, using block tree updates, satisfy*

$$\begin{aligned} J(\theta) &\stackrel{1}{=} \sum_T \rho(T) J(\theta^T) \geq \sum_T \rho(T) \max_{\mathbf{x}} \theta^T(\mathbf{x}) \\ &\stackrel{2}{=} \sum_T \rho(T) J(\hat{\theta}^T) \stackrel{3}{\geq} J(\hat{\theta}) \end{aligned} \tag{12}$$

where $\hat{\theta} = \sum_T \rho(T)\hat{\theta}^T \in F_L(\theta)$. *Equality in Step 3 would correspond to achieving weak tree agreement (shared maximizing assignment).*

Proof. The first equality $J(\theta) = \sum_T \rho(T) J(\theta^T)$ follows directly by substitution. Each $J(\theta^T) \geq \max_{\mathbf{x}} \theta^T(\mathbf{x})$ since $J(\theta^T)$, for $\theta^T \in F_L(\theta^T)$, is a dual LP relaxation and therefore its value upper bounds the corresponding MAP value for the tree T with parameters θ^T. The pairwise LP relaxation is exact for any tree and therefore the dual objective attains the MAP value $\max_{\mathbf{x}} \theta^T(\mathbf{x}) = \min_{\theta \in F_L(\theta^T)} J(\theta) = J(\hat{\theta}^T)$. $J(\theta)$ is a convex function as a point-wise maximum of potentials and therefore the last inequality corresponds to Jensen's inequality. The Jensen's inequality is tight only if all the tree models being averaged have at least one common maximizing local assignment for each pairwise and single node potential terms. This is weak tree agreement. The fact that $\hat{\theta} \in F_L(\theta)$ follows from Eq. (11) and because the block tree updates return a $\hat{\theta}^T$ that is a reparameterization of θ^T. □

A monotone version of the algorithm, known as TRW-S, was introduced by Kolmogorov (2006). One key difference is that in Step 3, only one node or one edge is averaged, and then max-product is run again on each tree. TRW-S is monotone in $\sum_T \rho(T) \max_{\mathbf{x}} \theta^T(\mathbf{x})$, but may not be for $J(\theta)$, depending on the reparameterization used. By using a slightly different weighting scheme, the algorithm in Figure 1 can be used to give an optimal reparameterization where one edge st has $\hat{\theta}^T_{st}(x_s, x_t) = \log \mu^T_{st}(x_s, x_t)$ (analogously for one node). Both TRW-S and this algorithm give the same solution for $\hat{\theta}_{st}(x_s, x_t)$. However, TRW-S would stop here, while our algorithm also averages over the other edges, obtaining a possibly larger (and never smaller) decrease in the dual objective. When the trees are monotonic chains, Kolmogorov (2006) shows how TRW-S can be implemented much more efficiently.

We observed empirically that using the tree block update algorithms sequentially to solve $F_L(\theta)$ converges more quickly than using them in parallel and averaging. However, the modified TRW algorithm is ideally suited for parallel processing, allowing us in Step 2 to independently find the optimal reparameterizations for each tree. By modifying the particular choice of trees, reparameterizations, and averaging steps, this framework could be used to derive various new parallel coordinate descent algorithms.

5 Transformations

We now relate each of the dual LPs in a monotone fashion, showing constructively that we can move from one representation to another while not increasing the common objective $J(f)$. We already showed an example of this in the previous section for the TRW dual. One of our goals is to clarify the relationship between the different dual formulations and algorithms. Some algorithms, such as the tree block update presented in Section 4, can be conveniently formulated for one of the representations, but because of these results, are applicable to the others as well. In addition, we could smoothly transition between the different representations throughout the optimization of the dual.

These transformations are easiest to illustrate by referring to the messages $\delta = \{\delta_{ji}(x_i)\}$ used in the definitions of each of the classes $F_L(\theta)$, $F_{L,E}(\theta)$, and $F_{L,V}(\theta)$. Recall that $F_L(\theta)$ puts no constraints on the messages δ, while $F_{L,E}$ requires that $\sum_{j \in N(i)} \delta_{ji}(x_i) = 0$, and $F_{L,V}$ requires that $\delta_{ji}(x_i) + \delta_{ij}(x_j) \geq \theta_{ij}(x_i, x_j)$.

The same messages, when used to construct potentials for two different classes (assuming the messages satisfy the constraints for both of these classes), can be used to identify members of both classes. However, the potentials will be different. For example, $f_\delta \in F_{L,V}(\theta)$ has pairwise terms identically zero, while the pairwise terms of $f'_\delta \in F_L(\theta)$ are of the form $f'_{ij}(x_i, x_j) = \theta_{ij}(x_i, x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j)$.

The transformations are specified in the following propositions, and are illustrated in Figure 3. The transformations resemble the sequential updates found in various message passing algorithms, but are generally weaker in terms of the effect on the objective. We begin with a simple transformation that removes the single node functions $f_i(x_i)$.

Proposition 5.1. *Consider any* $f_\delta \in F_L(\theta)$ *with messages* $\delta_{ji}(x_i)$. *Then* $f'_{\delta'} \in F_{L,E}(\theta)$, *defined by messages*

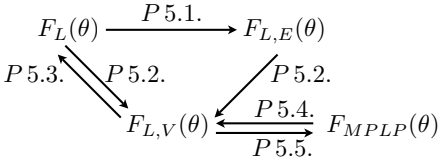

Figure 3: Monotone transformations between different representations.

$$\delta'_{ji}(x_i) = \delta_{ji}(x_i) - \frac{1}{|N(i)|}\sum_{k\in N(i)} \delta_{ki}(x_i) \tag{13}$$

satisfies $J(f_\delta) \geq J(f'_{\delta'})$.

Proof. The constraint $\sum_{j\in N(i)} \delta'_{ji}(x_i) = 0$ is satisfied. The transformation is monotone because

$$J(f_\delta) = \sum_i |N(i)| \max_{x_i} \frac{f_i^\delta(x_i)}{|N(i)|} + \sum_{ij\in E} \max_{x_i,x_j} f_{ij}^\delta(x_i,x_j)$$

$$\geq \sum_{ij\in E} \max_{x_i,x_j} \left\{ \frac{f_i^\delta(x_i)}{|N(i)|} + \frac{f_j^\delta(x_j)}{|N(j)|} + f_{ij}^\delta(x_i,x_j) \right\} = J(f'_{\delta'})$$

where we split up the node potentials, then combined maximizations (monotonic by convexity of max). □

We can also push all the information into the single node terms, effectively removing the edge functions $f_{ij}(x_i, x_j)$, as shown below.

Proposition 5.2. *Consider any* $f_\delta \in F_L(\theta)$ *or* $f_\delta \in F_{L,E}(\theta)$ *with messages* $\delta_{ji}(x_i)$. *Then* $f'_{\delta'} \in F_{L,V}(\theta)$, *defined by*

$$\delta'_{ji}(x_i) = \frac{1}{2}\delta_{ji}(x_i) + \frac{1}{2}\max_{x_j} \left\{\theta_{ij}(x_i,x_j) - \delta_{ij}(x_j)\right\} \tag{14}$$

satisfies $J(f_\delta) \geq J(f'_{\delta'})$.

Proof. We first show that δ' satisfies the constraint for $F_{L,V}(\theta)$. Since for any $\hat{x}_j$

$$\delta'_{ji}(x_i) \geq \frac{1}{2}\delta_{ji}(x_i) + \frac{1}{2}\theta_{ij}(x_i,\hat{x}_j) - \frac{1}{2}\delta_{ij}(\hat{x}_j),$$

we get by summing that $\delta'_{ji}(\hat{x}_i) + \delta'_{ij}(\hat{x}_j) \geq \theta_{ij}(\hat{x}_i,\hat{x}_j)$, as desired. To show that $J(f_\delta) \geq J(f'_{\delta'})$, first define $g_i^\delta(x_i)$ as

$$\sum_{j\in N(i)} \left(\frac{1}{2}\max_{x_j}\left\{\theta_{ij}(x_i,x_j) - \delta_{ij}(x_j)\right\} - \frac{1}{2}\delta_{ji}(x_i) \right).$$

We then split the edge potential in two, giving

$$\begin{aligned} J(f_\delta) &= \sum_i \max_{x_i} f_i^\delta(x_i) + \sum_i \sum_{j\in N(i)} \frac{1}{2}\max_{x_i,x_j} f_{ij}^\delta(x_i,x_j) \\ &\geq \sum_i \max_{x_i} f_i^\delta(x_i) + \sum_i \max_{x_i} g_i^\delta(x_i) \\ &\geq \sum_i \max_{x_i} \left\{ f_i^\delta(x_i) + g_i^\delta(x_i)\right\} = J(f_{\delta'}). \end{aligned}$$

Proposition 5.3. *Consider any* $f_\delta \in F_{L,V}(\theta)$ *with messages* $\delta_{ji}(x_i)$. *Then* $f'_\delta \in F_L(\theta)$ *defined in terms of the same messages* δ, *now also modifying edges, satisfies* $J(f_\delta) \geq J(f'_\delta)$.

Proof. The old messages must satisfy the constraint for $F_{L,V}(\theta)$, that $\theta_{ij}(x_i,x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j) \leq 0$. Thus, the new edge terms for $f'_\delta \in F_L(\theta)$ are all ≤ 0, so maximizing over them only decreases the objective. The node terms stay the same. □

While the transformation given in Proposition 5.3 may decrease the objective value, one can show that adding a constant to each message, in particular $\delta'_{ij}(x_j) =$

$$\delta_{ij}(x_j) + \frac{1}{2}\max_{x_i,x_j}\left\{\theta_{ij}(x_i,x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j)\right\}, \tag{15}$$

results in a $f'_{\delta'} \in F_L(\theta)$ such that $J(f_\delta) = J(f'_{\delta'})$. This now gives an *exact* mapping from $F_{L,V}(\theta)$, the dual given by Komodakis & Paragios (2008), to $F_L(\theta)$.

5.1 MPLP

A pairwise LP relaxation can also be obtained from the point of view of enforcing consistency in a directional manner, considering each edge in two different directions. The associated dual LP corresponds to dividing each edge potential into the associated nodes (Globerson & Jaakkola, 2008). More precisely, the objective is $\min_{f\in F_{MPLP}(\theta)} J(f)$, where the class $F_{MPLP}(\theta)$ is given by

$$\left\{ f : \begin{array}{l} f_i(x_i) = \sum_{j\in N(i)} \max_{x_j} \beta_{ji}(x_j,x_i) \\ f_{ij}(x_i,x_j) = 0 \\ \beta_{ji}(x_j,x_i) + \beta_{ij}(x_i,x_j) = \theta_{ij}(x_i,x_j) \end{array} \right\} \tag{16}$$

where each edge potential $\theta_{ij}(x_i,x_j)$ is divided into $\beta_{ji}(x_j,x_i)$ and $\beta_{ij}(x_i,x_j)$ for nodes i and j, respectively.

It is straightforward to show that $f_\beta \in F_{MPLP}(\theta)$ gives a valid sup-reparameterization similar to $F_{L,V}(\theta)$. The two formulations are indeed closely related. We show below how to move from one to the other.

Proposition 5.4. *Consider any* $f_\beta \in F_{MPLP}(\theta)$ *given by the dual variables* $\beta_{ji}(x_j,x_i)$. *Then* $f'_\delta \in F_{L,V}(\theta)$, *defined by*

$$\delta_{ji}(x_i) = \max_{x_j} \beta_{ji}(x_j,x_i) \tag{17}$$

satisfies $J(f_\beta) = J(f'_\delta)$.

Proof. The objectives are the same because $f_i^\beta(x_i) = f_i^\delta(x_i)$. Also, the constraint is satisfied, since $\delta_{ji}(\hat{x_i}) + \delta_{ij}(\hat{x_j}) \geq \beta_{ji}(\hat{x_j},\hat{x_i}) + \beta_{ij}(\hat{x_i},\hat{x_j}) = \theta_{ij}(\hat{x_i},\hat{x_j})$. □

Proposition 5.5. *Consider any $f_\delta \in F_{L,V}(\theta)$ given by the dual variables $\delta_{ji}(x_i)$. Then $f_\beta \in F_{MPLP}(\theta)$, defined by*

$$\beta_{ji}(x_j, x_i) = \theta_{ij}(x_i, x_j) - \delta_{ij}(x_j) \quad (18)$$
$$\beta_{ij}(x_i, x_j) = \delta_{ij}(x_j) \quad (19)$$

satisfies $J(f_\delta) \geq J(f_\beta)$.

Proof. For any $f_\delta \in F_{L,V}(\theta)$, $\delta_{ji}(x_i) + \delta_{ij}(x_j) \geq \theta_{ij}(x_i, x_j)$. Given our definition of $\beta_{ji}(x_j, x_i)$, $\delta_{ji}(x_i) \geq \max_{x_j} \beta_{ji}(x_j, x_i)$. Also, $\delta_{ij}(x_j) \geq \max_{x_i} \beta_{ij}(x_i, x_j)$ trivially. Therefore, for all i and x_i, $\sum_{j \in N(i)} \delta_{ji}(x_i) \geq \sum_{j \in N(i)} \max_{x_j} \beta_{ji}(x_j, x_i)$. □

6 Decodability

If the pairwise LP relaxation has a unique optimal solution and it is the MAP assignment, we could find it simply by solving the primal linear program. We use dual algorithms for reasons of *efficiency*. However, the dual solution is only useful if it helps us *find* the MAP assignment. In this section, we characterize when it is possible to decode the MAP assignment from the dual solution, using the common framework given by the dual $\min_{f \in F_L(\theta)} J(f)$. By using the transformations of the previous section, these results can be shown to apply to all of the algorithms discussed in this paper.

Duality in linear programming specifies *complementary slackness* conditions that every primal and dual optimal solution must satisfy. In particular, it can be shown that for any optimal μ^* for the pairwise LP relaxation given in Eqs. (2) and (3) and any optimal f^* for the dual $\min_{f \in F_L(\theta)} J(f)$,

$$\mu_i^*(\hat{x}_i) > 0 \;\Rightarrow\; f_i^*(\hat{x}_i) = \max_{x_i} f_i^*(x_i), \quad (20)$$
$$\mu_{ij}^*(\hat{x}_i, \hat{x}_j) > 0 \;\Rightarrow\; f_{ij}^*(\hat{x}_i, \hat{x}_j) = \max_{x_i, x_j} f_{ij}^*(x_i, x_j).$$

We will use these complementary slackness conditions to give conditions under which we can recover the MAP assignment from the dual optimal solution.

Definition 6.1. *We say that $f \in F_L(\theta)$* locally supports $\mathbf{x}^*$ *if $f_{ij}(x_i^*, x_j^*) \geq f_{ij}(x_i, x_j)$ for all $ij \in E$, x_i, x_j, and $f_i(x_i^*) \geq f_i(x_i)$ for all $i \in V, x_i$.*

Our claims refer to the pairwise LP relaxation being *tight* for some θ. By this, we mean that the dual value $\min_{f \in F_L(\theta)} J(f) = J(f^*)$ equals the MAP value. As a result, each MAP assignment (there can be more than one) represents a primal optimal solution. In addition, there may be fractional solutions that are also primal optimal, i.e., attain the MAP value.

Lemma 6.2. *When the pairwise LP relaxation is tight, every optimal $f^* \in F_L(\theta)$ locally supports every MAP assignment $\mathbf{x}^*$. Conversely, if any dual feasible $f \in F_L(\theta)$ supports an assignment $\mathbf{x}^*$, then f is optimal, the LP is tight, and $\mathbf{x}^*$ is a MAP assignment.*

Proof. The lemma is a simple consequence of complementary slackness. To show the first part, apply Eq. (20) to each MAP assignment $\mathbf{x}^*$. Since $\mu_{ij}^*(x_i^*, x_j^*) = 1$ for the corresponding primal solution, x_i^*, x_j^* must maximize $f_{ij}^*(x_i, x_j)$. The second part follows from the fact any primal solution that $f \in F_L(\theta)$ supports attains the same value. □

Lemma 6.2 is closely related to decodability results for convex max-product BP (Weiss *et al.*, 2007). The beliefs at the fixed-points of convex max-product BP can be shown to give a dual feasible (not necessarily optimal) $f \in F_L(\theta)$ with the property that, if the beliefs support an assignment, f does too. Thus, this must be a MAP assignment. Our result also characterizes when it is possible to find the MAP assignment with convex max-product BP by looking for supporting assignments: only when the LP relaxation is tight.

The search for a locally supporting assignment may be formulated as a satisfiability problem, satisfiable only when the LP is tight. If the variables are binary, the corresponding 2SAT problem can be solved in linear time (Johnson, 2008). However, when some variables are non-binary, finding a satisfying assignment may be intractable. We now look at a setting where reading off the solution from f^* is indeed straightforward.

Definition 6.3. *We say that $f \in F_L(\theta)$ is node* locally decodable *to $\mathbf{x}^*$ if $f_i(x_i^*) > f_i(x_i)$ for all $i, x_i \neq x_i^*$.*

The definition for edge local decodability is analogous. If solving the dual problem results in a locally decodable solution, then we can easily construct the MAP assignment from each node or edge (cf. Lemma 6.2). However, in many cases this cannot happen.

Lemma 6.4. *A dual optimal $f^* \in F_L(\theta)$ can be node or edge locally decodable only if the MAP assignment is unique and the pairwise LP is tight.*

Proof. Either node or edge local decodability suffices to uniquely determine a supporting assignment. If any dual feasible $f \in F_L(\theta)$ supports an assignment, then the assignment attains the dual value, thus the LP must be tight. When the LP is tight, the optimal f^* has to support all the MAP assignments by Lemma 6.2. Thus f^* can be locally decodable only if the MAP assignment is unique. □

But, how can we find a locally decodable solution when one exists? If the MAP assignment $\mathbf{x}^*$ is unique, then evaluating max-marginals is one way to get a locally

decodable solution $f^* \in F(\theta)$. Under some conditions, this holds for a dual optimal $f^* \in F_L(\theta)$ as well.

Theorem 6.5. *Assume the MAP assignment* $\mathbf{x}^*$ *is unique. Then,*

1. *if the pairwise LP is tight and has a unique solution, there exists* $f^* \in F_L(\theta)$ *that is locally decodable to* $\mathbf{x}^*$.
2. *for a tree structured model, the tree block updates given in Section 4 construct* $f^* \in F_L(\theta)$ *which is node locally decodable.*

Proof. (Sketch). The first claim follows from strict complementary slackness (Vanderbei, 2007). We can always find a primal-dual pair (μ^*, f^*) that satisfies the implication in Eq. (20) both ways. Since μ^* is unique, it corresponds to the unique MAP assignment $\mathbf{x}^*$, and the strict complementary slackness guarantees that $f^* \in F_L(\theta)$ is locally decodable.

The second claim trivially holds for the max-product tree block update since each $f_i^{(t+1)}(x_i)$ is given by the single node max-marginals and MAP is unique.

We now show the second claim for the sequential algorithm. Assume without loss of generality that the node potentials $f_i(x_i) = 0$. The block tree update contracts a tree into an edge by propagating max-marginals towards edge ij. Let $\hat{\theta}_{ij}(x_i, x_j)$ be $\sum_{k \in N(i) \setminus j} m_{k \to i}(x_i) + f_{ij}(x_i, x_j) + \sum_{k \in N(j) \setminus i} m_{k \to j}(x_j)$. Since the MAP assignment is unique, $\hat{\theta}_{ij}(x_i, x_j)$ must have the unique maximizer x_i^*, x_j^*. The edge reparameterization sets $\delta_{ij}(x_j)$ and $\delta_{ji}(x_i)$ so that the updated single node term $\hat{\theta}_i(x_i) = \sum_{k \in N(i) \setminus j} m_{k \to i}(x_i) + \delta_{ji}(x_i) \propto \max_{x_j} \hat{\theta}_{ij}(x_i, x_j)$. Thus, $\hat{\theta}_i(x_i)$ uniquely decodes to x_i^*.

Next we show that the subtrees associated with i and j, after setting $\delta_{ji}(x_i)$ and $\delta_{ij}(x_j)$, also uniquely decode to $\mathbf{x}^*$. The subtree rooted at i has a max-marginal of $\hat{\theta}_i(x_i)$ for node i. Thus, the MAP assignment for this subtree must have $x_i = x_i^*$. The remaining variables' maximizers are independent of how we set $\delta_{ji}(x_i)$ once the assignment to x_i is fixed, and so must also be maximized at $\mathbf{x}^*$. We can now apply this argument recursively. After the last edge incident on node i is updated, $\hat{\theta}_i(x_i)$ equals $f_i^{(t+1)}(x_i)$, maximized at x_i^*. □

A slightly different tree block update constructs a solution that is edge locally decodable. The above theorem shows that we can efficiently construct locally decodable dual solutions for trees. This could also be useful for non-tree models if repeated applications of the tree block updates move towards solutions that are locally decodable. An interesting open problem is to design algorithms that are guaranteed to return a solution that is locally decodable, for general graphs.

7 Conclusion

We have placed several dual formulations of the pairwise LP relaxation for MAP under a unified functional view. As a result, algorithms for these can be understood as optimizing a common objective, and analyzed theoretically as a group.

There are a number of immediate generalizations of this work. For example, the generalization to non-pairwise models is straightforward. Also, if the pairwise LP relaxation is not tight, we may wish to include higher-order cluster consistency constraints. The functional characterization can be extended to this setting, with similar equivalence transformations as presented here. The tree-block update scheme would then be given for hyper-trees.

Cycles are typically difficult to fully optimize using local block coordinate-descent algorithms. We believe that efficient block updates, similar to those given in this paper, can be derived directly for cycles instead of trees. Finally, much of our work here contributes to inference problems involving marginals as well.

Acknowledgements

The work was supported in part by the DARPA Transfer Learning Program. We thank Amir Globerson and Daphne Koller for helpful discussions, and Amir for his feedback on the paper.

References

Globerson, A., & Jaakkola, T. 2008. Fixing Max-Product: Convergent Message Passing Algorithms for MAP LP-Relaxations. *In: NIPS 21.*

Johnson, J. 2008. *Convex Relaxation Methods for Graphical Models: Lagrangian and Maximum Entropy Approaches.* Ph.D. thesis, EECS, MIT.

Kolmogorov, V. 2006. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(10), 1568–1583.

Komodakis, N., & Paragios, N. 2008. Beyond Loose LP-Relaxations: Optimizing MRFs by Repairing Cycles. *Pages 806–820 of: ECCV.*

Vanderbei, R.J. 2007. *Linear Programming: Foundations and Extensions.* 3rd edn. Springer.

Wainwright, M., Jaakkola, T., & Willsky, A. 2005. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Trans. on Information Theory*, **51**(11), 3697–3717.

Weiss, Y., Yanover, C., & Meltzer, T. 2007. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. *In: UAI.*

Werner, T. 2007. A Linear Programming Approach to Max-Sum Problem: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(7), 1165–1179.