Accelerated dual decomposition for MAP inference

Vladimir Jojic VJOJIC@CS.STANFORD.EDU
Stephen Gould SGOULD@STANFORD.EDU
Daphne Koller KOLLER@CS.STANFORD.EDU
Computer Science Department, 353 Serra Mall, Stanford University, Stanford, CA 94305, USA

Abstract

Approximate MAP inference in graphical models is an important and challenging problem for many domains including computer vision, computational biology and natural language understanding. Current state-of-the-art approaches employ convex relaxations of these problems as surrogate objectives, but only provide weak running time guarantees. In this paper, we develop an approximate inference algorithm that is both efficient and has strong theoretical guarantees. Specifically, our algorithm is guaranteed to converge to an ϵ-accurate solution of the convex relaxation in $O\left(\frac{1}{\epsilon}\right)$ time. We demonstrate our approach on synthetic and real-world problems and show that it outperforms current state-of-the-art techniques.

1. Introduction

Markov random fields (MRFs) are a useful framework for modeling many important problems in computer vision, computational biology and natural language understanding. One important task for these problems is to find the assignment to each variable that jointly minimizes the energy (maximizes the probability) defined by the model. However, finding the optimal solution for a general MRF is an NP-hard problem. Hence, research has focused on approximate methods. These methods, however, still pose computational challenges, and algorithms with fast convergence rates to near-optimal solutions are needed. This is especially important when inference is used as the inner loop of another algorithm, such as the popular max-margin learning algorithms (Tsochantaridis et al., 2005; Taskar et al., 2005), which iteratively add constraints to the training objective. Learning in these methods is significantly improved with better quality inference (Finley & Joachims, 2008).

Current state-of-the-art methods for approximate inference employ convex programming relaxations to find an approximate solution via the convex dual (Kolmogorov, 2006; Globerson & Jaakkola, 2007; Wainwright et al., 2005; Komodakis & Paragios, 2009). The solution to the original *primal* problem is found by a decoding process on the dual variables. Typically, the better the solution to the dual, the better the quality of the solution to the original problem. While many state-of-the-art approaches aim to optimize the dual formulation, they are either slow to converge or can get stuck (Globerson & Jaakkola, 2007; Kolmogorov, 2006). For example, coordinate descent procedures are prone to this failure mode. In contrast to previous methods, the method of Komodakis & Paragios (2009) provided guarantees on quality of convergence. However, since this is a projected gradient approach the guaranteed rate of convergence is slow, namely $O(\frac{1}{\epsilon^2})$ time complexity for an ϵ-accurate solution.

In this paper, we develop an algorithm based on the smoothing approach of Nesterov (2005) with significantly faster time complexity of $O(\frac{1}{\epsilon})$. Like the method of Komodakis & Paragios (2009), our method is based on the framework of Lagrangian relaxation that leverages the structure of the problem. Lagrangian relaxation divides the original problem into a set of coordinated *slave* problems that have tractable structure. The coordination of the slaves is handled by a *master*. The master achieves this by collecting *messages* from each of the slaves, which describe their solutions. Each slave is then updated in a manner so as to bring them into agreement. An important feature of the framework is that the tractability of the slaves is unaffected by the message passing. Note that, since the slave problems are decoupled, the algorithm can be distributed and the computational effort involved in finding their solution is fixed across all iterations.

Unlike projected subgradient approaches, we construct

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

smooth approximations to the slave problems that remain tractable and enable globally faster optimization algorithms to be applied. Importantly, our approximations are uniformly bounded, resulting in guaranteed convergence to an ϵ-accurate solution of the original problem.

Surprisingly, our algorithm, which is aimed at solving the MAP problem, gives rise to sum-product-like updates at a particular temperature. Such connections between max-product and sum-product algorithms have been explored in the context of zero-limit temperatures (Meltzer et al., 2009). To our knowledge, this is the first constructive approach to establishing such relationships at non-zero temperatures.

While the smoothed objective can be solved using any standard gradient-based approach (conjugate gradients, L-BFGS, etc.), we derived a method from a family of optimal methods (Nesterov, 1983), which provide superior convergence guarantees. Specifically, our method converges to an ϵ-approximate solution in $O(\frac{1}{\epsilon})$ iterations. In addition to the strong theoretical guarantees, we provide experimental results on both synthetic and real-world problems demonstrating the superiority of our approach.

2. Related Work

Our work stems from Lagrangian relaxation of the MAP estimation problem. This approach, which has been explored by Komodakis et al. (2007), leverages the tractability of the slave problems. The approach derives a dual-decomposition algorithm that only requires MAP inference in the slave. Although the algorithm is provably convergent, that is, it does not get stuck in local minima, the approach suffers from poor run time guarantees. In particular, it can take up to $O\left(\frac{1}{\epsilon^2}\right)$ iterations to converge. Our algorithm, on the other hand, converges in $O(\frac{1}{\epsilon})$ iterations.

The tree-reweighted (TRW) family of algorithms (Wainwright et al., 2005; Kolmogorov, 2006) are based on dual-decomposition across spanning trees. However, unlike (Komodakis & Paragios, 2009), these algorithms do not provide convergence guarantees.

The innovative approach of Globerson & Jaakkola (2007) derives a message-passing algorithm which iteratively optimizes the dual of the relaxed integer program. This is accomplished by block coordinate descent moves; thus the algorithm, due to its coordinate nature, can get stuck. Furthermore, the method has no convergence guarantees, but seems to perform well in practice.

Johnson et al. (2007) investigated smoothing of Lagrangian relaxations and utilized a gradient descent algorithm for their optimization. However, this work does not provide guarantees on either run time or on the level of error introduced by their smoothing approximation. In contrast to the above three approaches, our method provides strong convergence guarantees.

Ravikumar et al. (2008) introduced a family of double loop algorithms. The outer loop iterates over a schedule of smoothing coefficients and the inner loop solves an equivalent of a sequential TRW problem for a given smoothing coefficient. Strong guarantees exist; however these do not apply when the inner loop cannot be solved exactly. Further, solution of the inner loop requires sequential updates across different tree subproblems, eliminating one of the main sources of parallelism inherent in the dual decomposition based methods. In contrast, we provide strong guarantees on *total* running time, and each slave is solved exactly. Additionally, in our algorithm, slave problems can be solved in isolation, thus leveraging large scale distribution on top of an efficient algorithm.

3. Background

3.1. Markov Random Fields

Let $\mathbf{z} = (z_1, \ldots, z_N)$ be a vector of discrete random variables, each random variable z_i taking its value from a discrete set $\mathcal{Z}_i$. A Boltzmann distribution induced by energy function $E(\mathbf{z})$ over variables $\mathbf{z}$ at temperature μ is defined by $p(\mathbf{z}) = \frac{1}{Z} \exp\left\{-\frac{1}{\mu}E(\mathbf{z})\right\}$ where $Z = \sum_{\mathbf{z}} \exp\left\{-\frac{1}{\mu}E(\mathbf{z})\right\}$ is referred to as the partition function and serves to normalize the distribution. Markov random fields (MRFs) encode such a distribution and their energy function is formed as a sum of different potentials $E(\mathbf{z}) = -\sum_{c=1}^{C} \phi_c(\mathbf{z}_{\pi_c})$ at a temperature $\mu = 1$. Here, we assume that each potential ϕ_c is defined over a subset of the variables $\pi_c \subseteq \{1, \ldots, N\}$. We introduce $\pi_i^{-1} = \{c : i \in \pi_c\}$ to represent the set of potentials with variable z_i in their scope. In what follows, we will assume that inference is tractable for each potential; however, we make no assumption about the structure of the potential. Tractability can be achieved through exploitation of the structure of the potential (e.g., trees or low tree-width subgraphs) or the small size of its configuration space, i.e., $|\mathcal{Z}_{\pi_c}| = \prod_{i \in \pi_c} |\mathcal{Z}_i|$.

A typical task in MRF inference is to find the most likely joint assignment $\mathbf{z}^\star$ under the distribution $p(\mathbf{z})$, explicitly $\mathbf{z}^\star = \operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}) = \operatorname{argmax}_{\mathbf{z}} \sum_c \phi_c(\mathbf{z}_{\pi_c})$.

This problem can be recast as the integer program

$$\begin{array}{ll} \underset{\mathbf{z}}{\text{maximize}} & \sum_{c=1}^{C} \phi_c(\mathbf{z}_{\pi_c}) \\ \text{subject to} & z_i \in \mathcal{Z}_i \quad i = 1, \ldots, N. \end{array} \tag{IP}$$

The tractability of this integer program is made hard by the coupling of potentials sharing the same variables. As we will see, a framework known as Lagrangian relaxation will enable us to take advantage of our ability to efficiently solve tractable subproblems (ϕ_c) and combine these sub-solutions into a solution of the main problem. Naturally, the relaxation of the integer program leads to an approximation, and Lagrangian relaxation bounds can be shown to be at least as tight as the linear programming relaxations (Wolsey, 1998).

3.2. Lagrangian relaxation

In order to facilitate derivation of the algorithm and elucidate the character of messages, we introduce an equivalent representation of the integer program defined above. For each value $a \in \mathcal{Z}_i$ we introduce a binary integer variable $x_{i,a}$, indicating whether z_i assumes the value a. We will use x_i to denote a binary integer vector $(x_{i,1}, \dots x_{i,n_i})$, and define set $\mathcal{X}_i = \{x_i : \sum_{a \in \mathcal{Z}_i} x_{i,a} = 1\}$. Hence there exists a one-to-one mapping of configurations of x and z, $\text{bip}(x) = z$ and $|\mathcal{X}_i| = |\mathcal{Z}_i|$. We will use shorthand $\phi_c(x)$ for $\phi_c(\text{bip}(x))$ and let $\mathcal{X}_{\pi_c} = \text{bip}^{-1}(\mathcal{Z}_{\pi_c})$.

A binary integer program equivalent to (IP) is

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & \sum_c \phi_c(\mathbf{x}_{\pi_c}) \\ \text{subject to} & x_i \in \mathcal{X}_i \quad i = \{1, \dots, N\}. \end{array} \tag{BIP}$$

We obtain the Lagrangian relaxation of (BIP) via the process of dual decomposition (Bertsekas, 1999). First, we introduce copies of the original variables specific to each potential, with the corresponding potential indicated by the variables' superscript. These copies, slave variables, are constrained to agree with the original, master variables. The resulting equivalent integer program is

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{maximize}} & \sum_c \phi_c(\mathbf{x}^c_{\pi_c}) & \\ \text{subject to} & x^c_i \in \mathcal{X}_i & \forall c, i \in \pi_c \\ & x^c_{i,a} = x_{i,a} & \forall c, i \in \pi_c, a \in \mathcal{Z}_i \end{array}$$

It is important to note that the equality constraints between slave and master variables separate across the domain of the variables. We note that the domain of the x variables is heavily constrained, in that for any i only one of the indicators $x_{i,a}$ can be one. We introduce a partial Lagrangian

$$J(\nu, x, x^1, \dots x^C) = \sum_c \phi_c(\mathbf{x}^c_{\pi_c}) + \left\langle \nu^c_{\pi_c}, x^c_{\pi_c} - x_{\pi_c} \right\rangle.$$

The multiplier vector $\nu^c_{\pi_c}$ is of length equal to the total number of indicator variables $x^c_{i,a}$ with $i \in \pi_c$, that is $\sum_{i \in \pi_c} |\mathcal{Z}_i|$. Maximization of the partial Lagrangian over x yields the dual

$$\begin{array}{ll} \underset{\nu}{\text{minimize}} & \sum_c \max_{x^c_{\pi_c}} \phi_c(\mathbf{x}^c_{\pi_c}) + \left\langle \nu^c_{\pi_c}, x^c_{\pi_c} \right\rangle \\ \text{subject to} & \sum_{c \in \pi_i^{-1}} \nu^c_{i,a} = 0 \quad \forall i, a \in \mathcal{Z}_i. \end{array} \tag{LR}$$

By virtue of being a dual, this problem is convex in ν. It is composed of a sum of disjoint maximization problems and is nonsmooth. We identify slave problems

$$s^c(\nu^c_{\pi_c}) = \max_{\mathbf{x}^c_{\pi_c} \in \mathcal{X}} \phi_c(\mathbf{x}^c_{\pi_c}) + \left\langle \nu^c_{\pi_c}, x^c_{\pi_c} \right\rangle. \tag{SLV}$$

Each slave problem is an integer program. The exact and MAP inference remain tractable in each slave as $\phi_c(x_{\pi_c})' = \phi_c(x_{\pi_c}) + \langle \nu_{\pi_c}, x_{\pi_c} \rangle$ retains the size and structure of $\phi_c(x_{\pi_c})$. The master problem is

$$\begin{array}{ll} \underset{\nu}{\text{minimize}} & m(\nu) = \sum_c s^c(\nu^c_{\pi_c}) \\ \text{subject to} & \sum_{c \in \pi_i^{-1}} \nu^c_{i,a} = 0, \quad \forall i, a \in \mathcal{Z}_i. \end{array} \tag{MSTR}$$

We draw attention to the constraint that ensures that the multipliers ν associated with variable z_i and its state a, sum to zero across all slave problems that have the variable z_i in its scope. Equally importantly, the slave problem s^c does not involve multipliers for variables that are not in the scope of ϕ_c. Thus we can conclude that the size of the slave's state space has not grown by the process of Lagrangian relaxation, and the cost of exact inference in the slave remains the same regardless of the messages.

Traditionally, Lagrangian relaxations are solved with a projected subgradient method, due to the nonsmoothness of the objective (Bertsekas, 1999; Held & Karp, 1970; Komodakis & Paragios, 2009). Such methods require specification of the step size schedule and are guaranteed to require at most $O(\frac{1}{\epsilon^2})$ iterations to achieve a cost of $f^t \leq f^\star + \epsilon$, where $f^\star$ is the optimum.

4. Accelerated optimization

In this section, we will construct a smooth approximation of the master problem's objective (MSTR). This construction will be accomplished by smoothing of the slave problems' objectives in Section 4.1. This section also will cover the quality of the approximation as well as exhibit the character of the gradients of the smoothed problem. In Section 4.2 we derive a general method for optimizing smooth objectives with constraints akin to the ones in the master problem. Section 4.3 specializes the algorithm from Section 4.2 to the smoothed master problem. The resulting method optimizes an approximate cost, but with a faster algorithm than the one used for the nonsmooth case. The approximation introduced by smoothing of the objective ends up being more than compensated for by the use of an optimal algorithm.

4.1. Smoothing the Lagrangian relaxation

Recently, a broad framework for smoothing structured convex objectives has been introduced (Nesterov, 2005). In this approach, an adjoint problem, in essence a dual problem, is constructed for a single nonsmooth component. The adjoint problem is then modified by the addition of a strongly concave term, thus introducing smoothness in the original problem. We will apply this general methodology to the problem (SLV) to smooth each of the slave problems.

A slave problem (SLV) can be rewritten as

$$\max_{q \in \Delta_c} \sum_{x_{\pi_c}} q(x_{\pi_c}) \left(\phi_c(x^c_{\pi_c}) + \left\langle \nu^c, x^c_{\pi_c} \right\rangle \right),$$

where Δ_c is the simplex over all configurations of x_{π_c}. We denote the dimension of Δ_c as $|\Delta_c|$. Consequently, q is a distribution over all configurations x_{π_c} and $|\Delta_c| = |\mathcal{X}_{\pi_c}|$. This reformulation has not changed the optimum of the problem. The optimal distributions q still put the probability mass only on configurations that maximize the original slave objective.

We now change the problem by adding a strongly concave contribution dependent on q

$$d^c(q) = \ln(|\Delta_c|) + \sum_{x_{\pi_c}} q(x_{\pi_c}) \ln q(x_{\pi_c})$$

This term is bound by 0 and $\ln(|\Delta_c|)$. It is strongly concave with concavity parameter 1 (Nesterov, 2005). This term can be seen as a proximal regularization that pulls the distributions q towards $q^c_0 \propto \mathbf{1}$, a uniform distribution across all configurations x_{π_c}. Addition of this proximal regularization term, weighted by μ, yields a smoothed slave problem

$$s^c_\mu(\nu^c) = \max_{q \in \Delta_c} \sum_{x_{\pi_c}} q(x_{\pi_c}) \left(\phi_c(x^c_{\pi_c}) + \left\langle \nu^c, x^c_{\pi_c} \right\rangle \right) - \mu \left(\ln(|\Delta_c|) + \sum_{x_{\pi_c}} q(x_{\pi_c}) \ln q(x_{\pi_c}) \right),$$

which after maximization over q yields

$$s^c_\mu(\nu^c) = \mu \ln \left(\frac{1}{|\Delta_c|} \sum_{x_{\pi_c}} \exp \left\{ \frac{\phi_c(x_{\pi_c}) + \left\langle \nu^c, x^c_{\pi_c} \right\rangle}{\mu} \right\} \right).$$

The smoothing of the slave integer problem produced an objective that is closely related to the log partition function of an MRF.

Proposition 4.1 *The gradient of a smooth slave problem $s^c_\mu(\nu^c)$ is given by*

$$\frac{\partial s^c_\mu(\nu^c)}{\partial \nu^c_{i,a}} = p^c_\mu(x^c_{i,a} = 1)$$

where $p^c_\mu(x_{\pi_c}) \propto \exp \left\{ \frac{1}{\mu} \left(\phi_c(x_{\pi_c}) + \left\langle \nu^c, x^c_{\pi_c} \right\rangle \right) \right\}$.

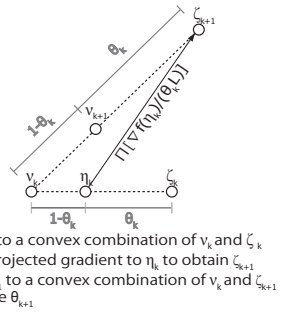

Figure 1. Schematic view of Algorithm 1

For proof see Appendix A.

With smooth slaves and their gradients in hand we can state the full smoothed master problem as

$$\begin{array}{ll} \underset{\nu}{\text{minimize}} & m_\mu(\nu) = \sum_c s^c_\mu(\nu^c) \\ \text{subject to} & \sum_c \nu_c = \mathbf{0}. \end{array} \qquad \text{(SMSTR)}$$

Proposition 4.2 *For any feasible ν*

$$m_\mu(\nu) \leq m(\nu) \leq m_\mu(\nu) + \mu \sum_c \ln |\Delta_c|.$$

For proof see Appendix A.

In addition to stating that we have obtained a uniform smooth approximation to the original nonsmooth objective, we point out that the choice of temperature μ and the sizes of slave-MRF configuration spaces dictate how close the smooth objective is to the original objective.

For the sake of completeness, we now state the partial derivatives of the master problem that will be used in the optimization algorithm:

$$\frac{\partial m_\mu(\nu^c)}{\partial \nu^c_{i,a}} = \frac{\partial s^c_\mu(\nu^c)}{\partial \nu^c_{i,a}} = p^c_\mu(x_{i,a} = 1).$$

4.2. An optimal method derivation

The chief benefit of working with a smooth objective is that its optimization can be performed with a variety of projected gradient based methods. Remarkably, there exist first-order methods that have guarantees of a better convergence rate than the naive projected gradients. Algorithms achieving the provably optimal error rate of $O(\frac{1}{k^2})$ in k iterations on problems with a known Lipschitz constant are called optimal methods (Nesterov, 1983). Here we construct an optimal method for problems akin to the smoothed master (SMSTR) problem constructed in Section 4.1. Given a differentiable and convex f, the problem of interest is:

$$\begin{array}{ll} \text{minimize} & f(\nu) \\ \text{subject to} & \sum_c \nu^c = 0. \end{array} \qquad \text{(P)}$$

Algorithm 1 An optimal method for smooth function optimization

1: **initialize** $\nu_k = \mathbf{0}$, $\eta_k = \mathbf{0}$, $\zeta_k = \mathbf{0}$, $\theta_0 = 1$
2: **for** $k \geq 0$ **do**
3: $\quad \eta_k = (1-\theta_k)\nu_k + \theta_k \zeta_k$
4: $\quad \zeta_{k+1} = \text{argmin}_{\sum_c \nu^c = 0}\{l_f(\nu;\eta_k) + \theta_k L D(\nu,\zeta_k)\}$
5: $\quad \nu_{k+1} = (1-\theta_k)\nu_k + \theta_k \zeta_{k+1}$
6: $\quad \theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
7: **end for**

We note that there exist optimal algorithms that retain all gradients from previous iterations (Nesterov, 2005). In large-scale MAP inference problems this is impractical. Hence we opt for a method that has limited memory requirements and achieves ϵ-solution in $O\left(\sqrt{\frac{L}{\epsilon}}\right)$ iterations. One such algorithm is Algorithm 1 (schematically shown in Figure 1). In order to fully specify the algorithm, we define l_f and proximal function D:

$$\begin{aligned} l_f(\nu;\eta) &:= f(\eta) + \langle \nabla f(\eta), \nu - \eta \rangle \\ D(\nu,\eta) &:= (1/2)\left\|\nu - \eta\right\|_2^2. \end{aligned}$$

Proposition 4.3 *For a differentiable and convex $f(\nu)$, where $\nu \in \mathbf{R}^N$, Algorithm 1 produces an ϵ-solution of the problem (P) in $O\left(\sqrt{\frac{L}{\epsilon}}\right)$ iterations, where L is the Lipschitz constant of function f.*

This proposition follows from Corollary 1 in (Tseng, 2008).

We note that for the update in line 4 of Algorithm 1 we need to compute

$$\begin{aligned} &\text{argmin}_{\sum_c \nu^c = 0} \quad \{l_f(\nu;\eta_k) + \theta_k L D(\nu,\zeta_k)\} = \\ &\text{argmin}_{\sum_c \nu^c = 0} \quad \{f(\eta_k) + \langle \nabla f(\eta_k), \nu - \eta_k \rangle + \\ &\qquad\qquad \theta_k (L/2)\left\|\nu - \zeta_k\right\|_2^2\} \end{aligned}$$

Writing the Lagrangian of the minimization problem:

$$f(\eta_k) + \langle \nabla f(\eta_k), \nu - \eta_k \rangle + \theta_k (L/2)\left\|\nu - \zeta_k\right\|_2^2 + \gamma^T \sum_c \nu^c,$$

and relevant KKT conditions:

$$\begin{aligned} \nabla f(\eta_k) + \theta_k L(\nu - \zeta_k) + \gamma &= 0 \\ \textstyle\sum_c \nu^c &= 0, \end{aligned}$$

we obtain $\gamma = \frac{1}{C}\sum_c (\theta_k L \zeta_k - \nabla f(\eta_k))$ and $\nu = \zeta_k - \frac{\nabla f(\eta_k) + \gamma}{\theta_k L}$.

We note that the memory requirement of this algorithm is equal to 3×size(ν).

4.3. Accelerated dual decomposition

The smooth function m_μ can be optimized by the method introduced in Section 4.2. A specialization of Algorithm 1 to the problem of solving the smooth master problem is given in Algorithm 2. In Section 4.1 we computed the gradients of the smooth master problem. It turned out that in order to compute gradients we need to perform marginalization of the slave-MRF at temperature μ given by

$$p_\mu^c(x_{\pi_c}) \propto \exp\left\{\frac{1}{\mu}\left(\phi_c(x_{\pi_c}) + \langle \nu^c, x_{\pi_c}\rangle\right)\right\}$$

and that the gradient of the master problem is given by

$$\frac{\partial m_\mu(\nu^c)}{\partial \nu_{i,a}^c} = \frac{\partial s_\mu^c(\nu^c)}{\partial \nu_{i,a}^c} = p_\mu^c(x_{i,a} = 1).$$

By design, the slave problems are tractable and these marginal distributions can be computed *exactly* in an amount of time that is independent of μ and remains constant across iterations. Importantly, the computation of the marginals of each slave (line 3. in Algorithm 2) can be carried out in parallel. Given desired precision ϵ and a chosen decomposition into slave problems, the temperature and Lipschitz constant (see Appendix A) are given by $\mu = \frac{\epsilon}{2\sum_c |\Delta_c|}$ and $L = \frac{2\sum_c \ln|\Delta_c|}{\epsilon}$.

Note that any hard constraints that limit the space of possible configurations of a slave will result in a smaller simplex Δ_c and consequently a smaller Lipschitz constant.

Algorithm 2 An optimal method for smoothed dual decomposition

1: **initialize** $\nu_k = \mathbf{0}$, $\eta_k = \mathbf{0}$, $\zeta_k = \mathbf{0}$, $\theta_0 = 1$
2: **for** $k \geq 0$ **do**
3: $\quad \eta_k = (1-\theta_k)\nu_k + \theta_k \zeta_k$
4: $\quad$ **foreach** c compute $p_\mu^c(x_{i,a})$ for $i \in \pi_c, a \in \mathcal{Z}_i$;
5: $\quad \zeta_{k+1} = \textbf{project}(p, \zeta_k, \eta_k)$
6: $\quad \nu_{k+1} = (1-\theta_k)\nu_k + \theta_k \zeta_{k+1}$
7: $\quad \theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
8: **end for**
9:
10: **function project**(p, ζ, η)
11: **for** $i = 1..N, a \in \mathcal{Z}_i$ **do**
12: $\quad \gamma_{i,a} = \frac{1}{|\pi_i^{-1}|}\sum_{c \in \pi_i^{-1}} \theta_k L \zeta_{i,a}^c - p_\mu^c(x_{i,a})$
13: $\quad$ **for** $c \in \pi_i^{-1}$ **do**
14: $\quad\quad \nu_{i,a}^c = \zeta_{i,a}^c - \frac{1}{\theta_k L}(p_\mu^c(x_{i,a}) + \gamma_{i,a})$
15: $\quad$ **end for**
16: **end for**
17: **return** ν

5. Convergence guarantees

The algorithm constructed in the previous section is applied to the smooth problem (SMSTR). However, the ultimate assessment of this method's performance

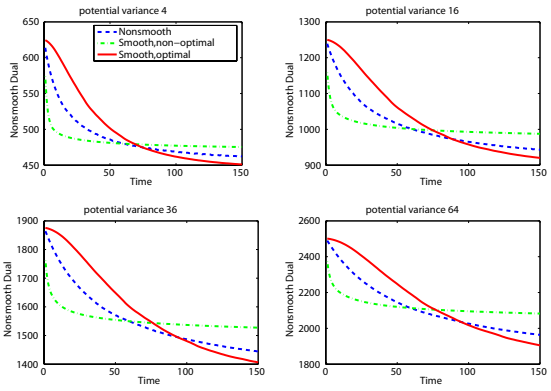

Figure 2. Synthetic Experimental Results. The MRF in all four examples is a grid of 30×30 random variables, each ranging over 7 states and involved in pairwise nearest neighbor potentials. These potentials are drawn from a zero mean Gaussian with variances ranging over $\{1, 4, 36, 64\}$. The three methods are: "Nonsmooth", projected subgradients applied to the nonsmooth objective; "Smooth, non-optimal", projected gradients applied to the smooth objective; "Smooth, optimal," the optimal method applied to the smooth objective. All solutions are evaluated in the non-smooth objective, and all solutions maintain dual feasibility in each iteration.

is how well it solves nonsmooth problem (MSTR). The following proposition answers this question.

Proposition 5.1 *For a given precision ϵ and a slave decomposition specified by the potentials ϕ_c, setting $\mu = \frac{\epsilon}{2\sum_c \ln|\Delta_c|}$, $L = \frac{1}{\mu}$ and iterating Algorithm 2 for $O\left(\sqrt{\frac{2L}{\epsilon}}\right)$ iterations produces solution $\hat{\nu}$ for which*

$$m(\hat{\nu}) \leq m(\nu^\star) + \epsilon,$$

where $\nu^\star$ is optimum of the nonsmooth problem (MSTR).

The proof of this proposition is given in Appendix A.

The projected subgradients on the original nonsmooth objective achieve this error in $O\left(\frac{1}{\epsilon^2}\right)$ iterations.

To summarize, we use a fast algorithm on an approximate cost to reach ϵ-solutions in the original cost faster than we could have done had we just optimized the original cost.

6. Experiments

6.1. Synthetic experiments

The synthetic experiments were performed on an MRF with pairwise nearest neighbor interactions forming a grid, and each random variable ranging over 7 states. The pairwise potentials were sampled from a Gaussian distribution with variance $\sigma^2 \in \{1, 4, 36, 64\}$. In all

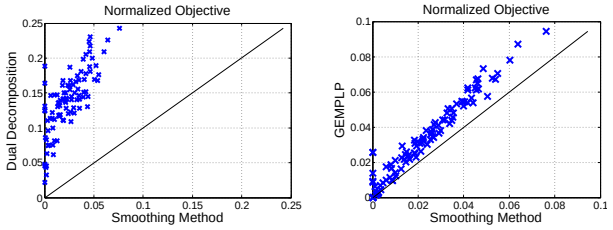

Figure 3. Comparison of our smooth-optimal method against (left) the dual decomposition method of Komodakis & Paragios (2009), and (right) LP relaxation algorithm of Globerson & Jaakkola (2007) on protein design dataset (Yanover et al., 2006).

tasks, we assume that the slaves are the faces of the grid (4-tuples of nodes and pairwise potentials on these nodes form a cycle). The potentials shared between multiple slaves are weighted by 0.5.

We compare three different methods. The first method optimizes the nonsmooth objective m using projected subgradients. The second method is gradient descent on the smoothed objective (m_μ). The third method is the one introduced in this paper, an optimal method on the smoothed objective (m_μ). The methods are compared in terms of the original nonsmooth objective m, the Lagrangian relaxation. The temperature μ is set so that the smooth objective is within $\epsilon = 1$ of the nonsmooth objective in all synthetic tests. The last two methods do not optimize this objective directly. However, by Proposition 4.2, the smooth objective is always within ϵ of the nonsmooth objective, thus limiting error due to optimization of the wrong objective. The projected gradients were used to solve a nonsmooth dual decomposition algorithm with a square summable step size ($\frac{1}{k}$ for k^{th} iteration).

Results are shown in Figure 2. The plotted time is the real running time of the algorithm rather than iterations of the methods. A single iteration of algorithms that optimize the smooth objective are the same; however, the projected subgradient method does not require computation of marginals. The difference in an iteration's running time is a result of the difference of the computational cost of performing `logsumexp` and `max` on a vector of values, which favors the projected subgradients method on the nonsmooth objective, but only as a multiplicative constant.

6.2. Protein Design

We conducted experiments on the protein design dataset made available by Yanover et al. (2006). The dataset consists of 97 problems with the goal of finding the most stable set of amino-acids and rotamer configurations that give rise to a given 3D configura-

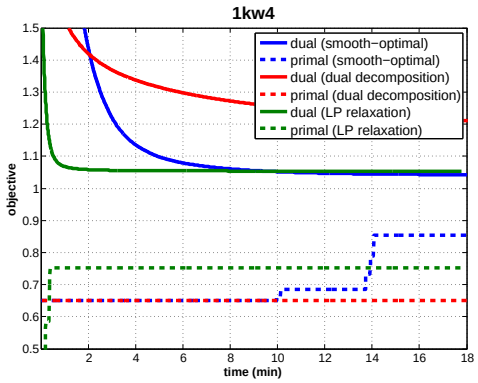

Figure 4. Typical optimization curves showing both the (original) dual objective and the decoded integer solution. Compared are our smooth-optimal method, the dual decomposition method of Komodakis & Paragios (2009) and the LP relaxation algorithm of Globerson & Jaakkola (2007).

tion. The problems are defined in terms of an energy function with unary and pairwise terms.

We compare our method to the dual decomposition method of Komodakis & Paragios (2009) and the LP-based coordinate-descent method of Globerson & Jaakkola (2007). For our method and dual decomposition, we absorb the unary potentials into the pairwise terms by distributing their contribution evenly across terms involving the same variable.[1] We set $\epsilon = 1$ for our method and run for a maximum of 5000 iterations on all methods.

Results for all 97 problems are shown in Figure 3. The plots show the final value of the (nonsmoothed) dual objective for our algorithm against the competing approaches. In the plots, a point above the diagonal indicates better performance. Our approach significantly outperforms the other methods.

We also show a plot of objective versus execution time for the three methods (see Figure 4). The plot shows both the non-smooth dual objective (solid lines) and the best integer primal solution at hand (dashed lines). The primal solution is determined by decoding the messages at each iteration (Globerson & Jaakkola, 2007). Notice that, initially, our objective decreases more slowly than the other approaches, but ultimately attains a better solution. Also observe that the best integer primal solution is only achieved after the dual objective is sufficiently close to its optimal. However, we note that monotonic decrease in the dual does not, in general, guarantee increase in the primal.

[1]That is, for unary potential $\phi_i(x_i)$ we add $\frac{1}{n_i}\phi_i(x_i)$ to each pairwise term $\phi_{ij}(x_i, x_j)$ where n_i is the number of pairwise terms involving variable x_i.

7. Discussion

In this paper we constructed a smooth approximation to the objective obtained from Lagrangian relaxation of the MAP inference problem. This enabled us to trade off quality of approximation in return for the ability to apply efficient gradient based methods. In particular, we used the "optimal method" of Nesterov (1983) which yielded a significantly faster algorithm compared to naive projected gradients. In addition, our construction revealed a connection between solving a MAP problem and sum-product-like algorithms at a particular low temperature. Ours is the first constructive illustration of such connections.

The smoothing of the objective also revealed the connection between the quality of the Lipschitz constant of this approximate objective and the form of the proximal term used in smoothing. Employing entropic proximal terms, as in our approach, is one of many possibilities. For example, the convex free energies studied by Meltzer et al. (2009) may offer alternative proximal terms that yield better, or even adaptive, Lipschitz constants.

Our experiments indicate that the initial iterations of our algorithms can be significantly accelerated by using an adaptive Lipschitz schedule. For example, such a schedule may be achieved through annealing of the smoothing coefficient. Regardless of the speedups obtained in initial iterations, the strength of our method stems from its strong guarantees of fast global convergence towards the optimal solution.

References

Bertsekas, D.P. *Nonlinear Programming*. Athena Scientific, 1999.

Finley, T. and Joachims, T. Training structural SVMs when exact inference is intractable. In *ICML*, 2008.

Globerson, A. and Jaakkola, T. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2007.

Held, M. and Karp, R. M. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

Johnson, J. K., Malioutov, D., and Willsky, A. S. Lagrangian relaxation for map estimation in graphical models. In *45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE PAMI*, 2006.

Komodakis, N. and Paragios, N. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009.

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

Meltzer, T., Globerson, A., and Weiss, Y. Convergent message passing algorithms - a unifying view. In *UAI*, 2009.

Nesterov, Yu. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet. Math. Dokl.*, 27:372–376, 1983.

Nesterov, Yu. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

Ravikumar, P. D., Agarwal, A., and Wainwright, M. J. Message-passing for graph-structured linear programs: proximal projections, convergence and rounding schemes. In *ICML*, pp. 800–807, 2008.

Taskar, B., Chatalbashev, V., Koller, D., and Guestrin, C. Learning structured prediction models: A large margin approach. In *ICML*, 2005.

Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Optimization Optim.*, 2008.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

Wainwright, M. J., Jaakkola, T., and Willsky, A. S. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Trans. on Info. Theory*, 2005.

Wolsey, L. A. *Integer Programming.* Wiley-Interscience, 1998.

Yanover, C., Meltzer, T., and Weiss, Y. Linear programming relaxations and belief propagation—an empirical study. *JMLR*, 2006.

A. Proofs of Propositions

Proof of Proposition 4.1. The slave problem is

$$s^c_\mu(\nu^c) = \max_{q\in\Delta_c} \quad \sum_{x_{\pi_c}} q(x_{\pi_c})\left(\phi_c(x^c_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right) - \mu\left(\ln(|\Delta_c|) + \sum_{x_{\pi_c}} q(x_{\pi_c})\ln q(x_{\pi_c})\right).$$

The partial derivatives with respect to $\nu^c_{i,a}$ of this smooth objective are

$$\frac{\partial s^c_\mu(\nu^c)}{\partial \nu^c_{i,a}} = \frac{\sum_{x_{\pi_c}} \exp\left\{\frac{1}{\mu}\left(\phi_c(x_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right)\right\} x^c_{i,a}}{\sum_{x_{\pi_c}} \exp\left\{\frac{1}{\mu}\left(\phi_c(x_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right)\right\}}.$$

We note that these quantities are exactly marginals of a MRF with potentials $\phi_c(x^c_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle$ at temperature μ, explicitly $p^c_\mu(x_{\pi_c}) \propto \exp\left\{\frac{1}{\mu}\left(\phi_c(x_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right)\right\}$. Hence we observe that $\frac{\partial s^c_\mu(\nu^c)}{\partial \nu^c_{i,a}} = p^c_\mu(x^c_{i,a} = 1)$.

Proof of Proposition 4.2. Since $d^c(q) = \ln(|\Delta_c|) - \sum_{x_{\pi_c}} q(x_{\pi_c}) \ln q(x_{\pi_c})$ is bounded by $0 \le d^c(q) \le \ln|\Delta_c|$, then

$$\begin{aligned} s^c_\mu(\nu^c) &= \max_{q\in\Delta_c} \sum_{x_{\pi_c}} q(x_{\pi_c})\left(\phi_c(x^c_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right) - \mu d^c(q) \\ &\ge s^c(\nu_c) - \max_{q\in\Delta_c} d^c(q) = s^c(\nu_c) - \mu\ln|\Delta_c| \\ s^c_\mu(\nu^c) &= \max_{q\in\Delta_c} \sum_{x_{\pi_c}} q(x_{\pi_c})\left(\phi_c(x^c_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right) - \mu d^c(q) \\ &\ge \max_{q\in\Delta_c} \sum_{x_{\pi_c}} q(x_{\pi_c})\left(\phi_c(x^c_{\pi_c}) + \langle \nu^c, x^c_{\pi_c}\rangle\right) = s^c(\nu^c) \end{aligned}$$

It follows that $s_\mu(\nu^c) \le s(\nu^c) \le s_\mu(\nu^c) + \mu\ln|\Delta_c|$ and hence $m_\mu(\nu) \le m(\nu) \le m_\mu(\nu) + \mu\sum_c \ln|\Delta_c|$.

Proof of Proposition 5.1. Assuming that $\nu^\star$ is the optimum of the nonsmooth problem (MSTR), ν^s is the optimum of the smooth problem (SMSTR) and $\hat\nu$ is the solution obtained after $O\left(\sqrt{\frac{2L}{\epsilon}}\right)$ iterations. The following inequality chain gives the desired bound on error

$$\begin{array}{rcll} m(\hat\nu) & \le & m_\mu(\hat\nu) + \frac{\epsilon}{2} & \text{(Prop. 4.2)} \\ & \le & m_\mu(\nu^s) + \epsilon & \text{(by algorithm rate)} \\ & \le & m_\mu(\nu^\star) + \epsilon & \text{(by optimality of } \nu^s\text{)} \\ & \le & m(\nu^\star) + \epsilon & \text{(Prop. 4.2)} \end{array}$$

Computing Lipschitz. In order to compute the Lipschitz constant for the master problem we rewrite the master problem as

$$\begin{aligned} m_\mu(\nu) &= \max_{q^1}\cdots\max_{q^C} \sum_c \left\langle q^c, \phi_c + A_c^T\nu^c\right\rangle - \mu d^c(q) \\ &= \max_{q^1}\cdots\max_{q^C} \langle q, \phi\rangle + \left\langle q, A^T\nu\right\rangle - \langle \mathbf{1}, \mu d(q)\rangle, \end{aligned}$$

where $q = [q^1; q^2; \cdots q^C]$ and $A = [A_1; A_2; \cdots A_C]$ and $d(q) = [d^1(q^1); d^2(q^2); \cdots d^C(q^C)]$. Further, each A_c is a binary matrix with $\sum_{i\in\pi_c}|\mathcal{X}_i|$ rows and $\prod_{i\in\pi_c}|\mathcal{X}_i|$ columns and $(A_c)_{(i,a),x} = x_{i,a}$. The Lipschitz constant is then $L = \frac{1}{\mu}\|A\|_{1,2}$ (Nesterov, 2005) with

$$\begin{aligned} &\|A\|_{1,2} = \max_{q,\nu}\{\langle \nu^T A, q\rangle : \|\nu\|_1 = 1, \|q\|_2 = 1\} \\ &= \max_\nu\left\{\max_q\left\{\left\|\langle \nu^T A, q\rangle\right\|_2 : \|q\|_2 = 1\right\} : \|\nu\|_1 = 1\right\} \\ &= \max_\nu\left\{\max_x\left\{|\nu^T(A_{\cdot,x})|\right\} : \|\nu\|_1 = 1\right\} \\ &= \max_{i,j}(|A|_{i,j}) = 1 \end{aligned}$$

Assuming both $\|\cdot\|_1$ and $\|\cdot\|_2$ are L1 norms $\|A\|_{1,2} = 1$.