Advanced Structured Prediction

Editors:

Sebastian Nowozin Sebastian.Nowozin@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

Peter V. Gehler pgehler@tuebingen.mpg.de
Max Planck Insitute for Intelligent Systems
72076 Tübingen, Germany

Jeremy Jancsary jermyj@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

Christoph Lampert chl@ist.ac.at
IST Austria
A-3400 Klosterneuburg, Austria

This is a draft version of the author chapter.

The MIT Press
Cambridge, Massachusetts
London, England

1 Smoothed Coordinate Descent for MAP Inference

Ofer Meshi meshi@ttic.edu
Toyota Technological Institute at Chicago
Chicago, IL

Tommi Jaakkola tommi@csail.mit.edu
CSAIL, MIT
Cambridge, MA

Amir Globerson gamir@cs.huji.ac.il
The Hebrew University
Jerusalem, Israel

Finding maximum a posteriori (MAP) assignments in graphical models is an important task in many applications. Since the problem is generally hard, linear programming (LP) relaxations are often used. Solving these relaxations efficiently is thus an important practical problem. In recent years, several authors have proposed message passing updates corresponding to coordinate descent in the dual LP. However, these are generally not guaranteed to converge to a global optimum. One approach to remedy this is to smooth the LP, and perform coordinate descent on the smoothed dual. Here we provide a tutorial introduction to such algorithms, followed by an analysis of their convergence rate. We analyze the rate of convergence to both the primal and dual optima of the problems, under different coordinate update schedules. Empirical evaluation supports our theoretical claims and shows that the method is highly competitive with state of the art approaches that yield global optima.

1.1 Introduction

Many applications involve simultaneous prediction of multiple variables. For example, we may seek to label pixels in an image with their semantic classes or find the semantic role of words in a sentence. These problems can be cast as maximizing a function over a set of labels (or minimizing an energy function). The function typically decomposes into a sum of local functions over overlapping subsets of variables.

Such maximization problems are nevertheless typically hard (Koller and Friedman, 2009). Even for simple decompositions (e.g., subsets correspond to pairs of variables), maximizing over the set of labels is often provably NP-hard. One approach would be to reduce the problem to a tractable one, e.g., by constraining the model to a low tree-width graph. However, empirically, using more complex interactions together with approximate inference methods is often advantageous. One popular family of approximate methods is the linear programming (LP) relaxation approach (e.g., see Chapter 8 in Wainwright and Jordan, 2008). Although these LPs are tractable, general purpose LP solvers typically do not exploit the problem structure (Yanover et al., 2006). Therefore a great deal of effort has gone into designing solvers that are specifically tailored to typical MAP-LP relaxations. These include, for example, cut based algorithms (Boykov et al., 1999; Kolmogorov and Rother, 2007), accelerated gradient methods (Jojic et al., 2010; Savchynskyy et al., 2011), and augmented Lagrangian methods (Martins et al., 2011; Meshi and Globerson, 2011).

One class of particularly simple algorithms, which we will focus on here, are coordinate minimization based approaches. Examples include max-sum-diffusion (Werner, 2007), MPLP (Globerson and Jaakkola, 2008) and TRW-S (Kolmogorov, 2006).[1] These work by first taking the dual of the LP and then optimizing the dual in a block coordinate fashion (see Sontag et al., 2011, for a review). In many cases, the coordinate block operations can be performed in closed form, resulting in updates quite similar to the max-product message passing algorithm. By coordinate minimization we mean that at each step a set of coordinates is chosen, all other coordinates are fixed, and the chosen coordinates are set to their optimal value given the rest. This is different from a coordinate descent strategy where instead a gradient step is performed on the chosen coordinates (rather than full optimization).

A main caveat of the coordinate minimization approach is that it will

1. See Chapter by Schoenemann and Kolmogorov in this volume for a generalized version of TRW-S

not necessarily find the global optimum of the LP (although in practice it often does). This is a direct result of the LP objective not being strictly convex. Several authors have proposed to smooth the LP with entropy terms and employ variants of coordinate minimization (Hazan and Shashua, 2010; Werner, 2009). However, the convergence rate of these methods has not been analyzed. Moreover, since the algorithms work in the dual, there is no simple procedure to map the result back into primal feasible variables. In this chapter, we provide a tutorial introduction to the smoothing approach to MAP LP relaxations, and analyze its convergence rate.

Convergence rates for coordinate minimization are typically hard to obtain. As mentioned above, convergence to the global optimum is in fact not always achieved, and in these cases the rate of convergence (e.g., to a local optimum) is not of much interest. However, there are many cases where global convergence is guaranteed asymptotically (e.g., Tseng, 2001) but the number of iterations needed to reach a given accuracy ϵ is not known.

The rate of convergence and its analysis depend on the update schedule used by the algorithm. Namely, the choice of which coordinate (or block of coordinates) to update at each step. The simplest schedule is to decide on a given fixed permutation of the coordinates and update those in a cyclic manner. However, this seems to be the hardest case to analyze, and only recently have results been obtained for convergence rate in specific problem settings (Saha and Tewari, 2013).

Choosing non-cyclic update schedules seems to considerably simplify the analysis. One variant is to randomly choose the next coordinate block. The convergence rate of this stochastic schedule has been analyzed recently in Nesterov (2010); Shalev-Shwartz and Tewari (2011). However, this was done not for coordinate minimization (which we study here) but rather for gradient descent steps along each coordinate. Another non-cyclic update is a greedy schedule, where at each step one chooses the coordinate that has the most "potential" for improving the objective.[2] Such greedy schemes have been studied for example in the context of coreset optimization and the Frank Wolfe method (Clarkson, 2010), but the resulting algorithms are different from coordinate minimization.[3]

In what follows, we review the smoothing approach to MAP LP relaxations, and analyze its convergence rate. We study both greedy and stochastic schedules and analyze their rate of convergence to the dual optimum. We also introduce a simple mapping to primal variables, and analyze the rate

2. This can be measured in various ways.
3. For example, they do not fully optimize each coordinate.

of convergence of these to the primal optimum.[4] Finally, we provide a brief review of the ADMM method, which illustrates a different way of using smoothing and coordinate descent to globally solve the MAP LP problem.

1.2 MAP and LP Relaxations

Consider a set of n discrete variables $X_1, \ldots, X_n$, and a set C of subsets of these variables (i.e., $c \in C$ is a subset of $\{1, \ldots, n\}$). We use x_i to denote particular assignments to these variables. Next, consider functions that decompose according to these subsets. In particular, each subset c is associated with a local function or factor $\theta_c(x_c)$ and we also include factors $\theta_i(x_i)$ for each individual variable.[5] These can be used to define the following "score" function, which returns a real value for any assignment to the n variables:

$$f(x_1, \ldots, x_n; \theta) = \sum_{c \in C} \theta_c(x_c) + \sum_{i=1}^{n} \theta_i(x_i) \ . \tag{1.1}$$

We will often write $f(x; \theta)$ where x corresponds to the assignment to the n variables.

The MAP problem is to find an assignment x to all the variables which maximizes $f(x; \theta)$. Namely:

$$\text{MAP}(\theta) = \max_{x_1, \ldots, x_n} f(x_1, \ldots, x_n; \theta) \ . \tag{1.2}$$

The above is a combinatorial optimization problem, whose naive solution requires searching over 2^n possible assignments (for binary variables). This is indeed the worst-case complexity of such problems.[6] Linear programming relaxations are a popular approach to approximating combinatorial optimization problems. In what follows we review the most common LP relaxation for the MAP problem (Wainwright and Jordan, 2008; Wainwright et al., 2005; Werner, 2007). See also Živný et al. and Schoenemann and Kolmogorov in this volume.

To obtain an LP relaxation for Eq. (1.2), we first write it as an integer linear program, and then relax the integrality constraints. Consider a set

4. A related analysis of MAP-LP using smoothing appeared in Burshtein (2009). However, their approach is specific to LDPC codes, and does not apply to general MAP problems as we analyze here.
5. Although singleton factors are not needed for generality, we keep them for notational convenience.
6. For example max-cut can be easily seen as an instance of Eq. (1.2).

of boolean variables $\mu_i(x_i) \in \{0,1\}$ (one for each $i \in \{1,\ldots,n\}$ and value of x_i), that are constrained to be either zero or one. A setting $\mu_i(x_i) = 1$ will reflect the fact that the i^{th} variable has the value x_i. Similarly, consider variables $\mu_c(x_c)$ (one for each $c \in C$ and value of x_c). A setting $\mu_c(x_c) = 1$ will reflect the fact that the variables in c are set to the value x_c. We denote an assignment to all the μ's by a vector μ

We can now cast the MAP problem as an integer linear program as follows:[7]

$$\begin{array}{ll} \max & \sum_c \sum_{x_c} \theta_c(x_c)\mu_c(x_c) + \sum_i \sum_{x_i} \theta_i(x_i)\mu_i(x_i) \\ s.t. & \sum_{x_{c\setminus i}} \mu_c(x_c) = \mu_i(x_i) \quad , \quad \forall c, i \in c, x_i \\ & \sum_{x_i} \mu_i(x_i) = 1 \quad , \quad \forall i \\ & \sum_{x_c} \mu_c(x_c) = 1 \quad , \quad \forall c \\ & \mu_i(x_i) \in \{0,1\} \quad , \quad \forall i, x_i \\ & \mu_c(x_c) \in \{0,1\} \quad , \quad \forall c, x_c \ , \end{array} \tag{1.3}$$

where in $\sum_{x_{c\setminus i}} \mu_c(x_c)$ we sum out all variables in c except i. To see that this is equivalent to the MAP problem, note that any feasible μ corresponds to an assignment $x_1,\ldots,x_n$.[8] The mapping is obtained by setting $X_i = x_i$ where x_i is the value such that $\mu_i(x_i) = 1$ (There is only one such value, because of the constraints). Denote the assignment corresponding to μ by $x(\mu)$. Then the objective in Eq. (1.3) is $f(x(\mu);\theta)$. Finally, since each assignment x has a corresponding μ we have the equivalence to MAP.

The above ILP can be naturally converted into an LP by relaxing the integrality constraint $\mu_\alpha(x_\alpha) \in \{0,1\}$ to $\mu_\alpha(x_\alpha) \in [0,1]$ (where α is either a variable i or a factor c). The resulting LP is:

$$\begin{aligned} PMAP: \max_{\mu \in \mathcal{M}_L} P(\mu) &= \max_{\mu \in \mathcal{M}_L} \left\{ \sum_c \sum_{x_c} \theta_c(x_c)\mu_c(x_c) + \sum_i \sum_{x_i} \theta_i(x_i)\mu_i(x_i) \right\} \\ &= \max_{\mu \in \mathcal{M}_L} \mu \cdot \theta \ , \end{aligned} \tag{1.4}$$

where $P(\mu)$ is the primal (linear) objective and $\mathcal{M}_L$ is given by:[9]

$$\mathcal{M}_L = \left\{ \mu \geq 0 : \begin{array}{ll} \sum_{x_{c\setminus i}} \mu_c(x_c) = \mu_i(x_i) & \forall c, i \in c, x_i \\ \sum_{x_i} \mu_i(x_i) = 1 & \forall i \end{array} \right\} \ . \tag{1.5}$$

The set $\mathcal{M}_L$ is often referred to as the *local marginal polytope* (Wainwright and Jordan, 2008).

---

7. Note that the normalization constraint on μ_c is redundant, but is usually included.
8. Our derivation here is similar to Weiss et al. (2007).
9. We can neglect the upper bound on μ, since it is enforced by the normalization constraint.

If the maximizer of $PMAP$ has only integral values, then it solves the ILP and $x(\mu)$ is the MAP solution. However, in the general case the solution may be fractional (Wainwright and Jordan, 2008) and the maximum of $PMAP$ is an upper bound on MAP(θ).

1.2.1 The Dual LP

The linear program in Eq. (1.4) can be solved in polynomial time using generic LP solvers. However, these do not use the special structure of the problem, and can often result in impractical running times (Yanover et al., 2006). Thus, in recent years considerable research effort has gone into designing specific algorithms for solving Eq. (1.4). Many of these are based on the notion of dual coordinate descent. Namely, they take the convex dual of Eq. (1.4) and perform block coordinate descent on its variables. Since our algorithms will be closely related to this approach, we briefly introduce the dual objective.

The dual variables will be denoted by $\delta_{ci}(x_i)$, where there is one such variable for each i, c, x_i. These can be intuitively understood as messages from subset c to node i, reflecting a belief that the i^{th} variable has value x_i. The convex dual of Eq. (1.4) is then:

$$\min_{\delta} \sum_{c} \max_{x_c} \left(\theta_c(x_c) - \sum_{i:i\in c} \delta_{ci}(x_i) \right) + \sum_{i} \max_{x_i} \left(\theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i) \right) . \quad (1.6)$$

This objective may also be derived using the dual decomposition framework (e.g., see Sontag et al., 2011; Komodakis et al., 2011).

The advantage of the optimization problem in Eq. (1.6) is that it is unconstrained, and thus can be optimized using simple convex optimization approaches. One example is sub-gradient descent and its accelerated variants (e.g., see Komodakis et al., 2011; Savchynskyy et al., 2011; Jojic et al., 2010).

Another nice property of Eq. (1.6) is that minimizing over some blocks of coordinates can be done in closed form given a fixed value of the other coordinates. This is the basis of the dual coordinate descent algorithms mentioned earlier. However, as also mentioned earlier, these algorithms are generally not guaranteed to converge to a global optimum. In what follows, we review a smoothing approach that preserves that nice structure of coordinate descent algorithms, but results in global convergence.

1.2.2 Smoothing the LP

Since global convergence is desirable, several authors have considered a smoothed version of the LP in Eq. (1.4). As we shall see, this offers several

advantages over solving the LP directly. The basic idea is as follows: given a parameter $\tau > 0$, we construct a new primal problem $PMAP_\tau$ that is $O(\frac{1}{\tau})$ *close* to the original $PMAP$ (see below for a precise definition). The dual of $PMAP$ is denoted by $DMAP_\tau$. It is very similar in structure to Eq. (1.6), but with one key difference: the global optimum of $DMAP_\tau$ can be found using coordinate descent, and with guarantees on convergence rate.

Define the following primal objective:

$$P_\tau(\mu) = \mu \cdot \theta + \frac{1}{\tau} \sum_c H(\mu_c) + \frac{1}{\tau} \sum_i H(\mu_i) \ , \tag{1.7}$$

where $H(\mu_c)$ and $H(\mu_i)$ are the entropies of the corresponding distributions.[10] Now define the following smoothed primal optimization problem:

$$PMAP_\tau : \quad \max_{\mu \in \mathcal{M}_L} P_\tau(\mu) \ . \tag{1.8}$$

Note that as $\tau \to \infty$ we obtain the original primal problem. In fact, a stronger result can be shown. Specifically, the optimal value of $PMAP$ is $O(\frac{1}{\tau})$ close to the optimal value of $PMAP_\tau$. This justifies using the smoothed objective P_τ as a proxy to P in Eq. (1.4). We express this in the following lemma (which appears in similar forms in Hazan and Shashua (2010); Nesterov (2005)).

Lemma 1.1. *Denote by μ^* the optimum of problem $PMAP$ in Eq. (1.4) and by $\hat{\mu}^*$ the optimum of problem $PMAP_\tau$ in Eq. (1.8). Then:*

$$\hat{\mu}^* \cdot \theta \leq \mu^* \cdot \theta \leq \hat{\mu}^* \cdot \theta + \frac{H_{\max}}{\tau} \ , \tag{1.9}$$

where $H_{\max} = \sum_c \log |x_c| + \sum_i \log |x_i|$ (here $|x_\alpha|$ is the number of possible configurations of variables or factors). In other words, the smoothed optimum is an $O(\frac{1}{\tau})$-optimal solution of the original non-smoothed problem.

1.2.3 The Dual of the Smoothed LP

As mentioned above, we shall be particularly interested in the dual of $PMAP_\tau$ since it facilitates simple coordinate minimization updates. As in the non smooth case (see Section 1.2.1), our dual variables will be denoted by $\delta_{ci}(x_i)$. Before introducing the dual objective, we define the *soft-max* function, which plays a key role in this dual. Given a function $v(x)$ over

10. Namely, $H(\mu_i) = -\sum_{x_i} \mu_i(x_i) \log \mu_i(x_i)$, and $H(\mu_c) = -\sum_{x_c} \mu_c(x_c) \log \mu_c(x_c)$.

some variable x, and a parameter τ, we define the soft-max of v by:

$$\underset{x}{\text{smax}}\,(v(x);\tau) = \frac{1}{\tau}\log\sum_{x}\exp\left(\tau v(x)\right) \ . \tag{1.10}$$

The soft-max is closely related to the max function in several ways:

- It upper bounds the max, namely: $\text{smax}_x\,(v(x);\tau) \geq \max_x v(x)$ for all τ values.
- As $\tau \to \infty$ the soft-max approaches the max. Namely:

$$\lim_{\tau\to\infty}\ \underset{x}{\text{smax}}\,(v(x);\tau) = \max_{x} v(x) \ . \tag{1.11}$$

We now turn to the dual of $PMAP_\tau$. Standard duality transformation show that the dual objective is (e.g., see Boyd and Vandenberghe, 2004):[11]

$$F(\delta) = \sum_{c}\underset{x_c}{\text{smax}}\left(\theta_c(x_c) - \sum_{i:i\in c}\delta_{ci}(x_i);\tau\right) + \sum_{i}\underset{x_i}{\text{smax}}\left(\theta_i(x_i) + \sum_{c:i\in c}\delta_{ci}(x_i);\tau\right) \ . \tag{1.12}$$

The dual is an unconstrained smooth minimization problem:

$$DMAP_\tau : \min_{\delta} F(\delta) \ . \tag{1.13}$$

Convex duality implies that the optima of $DMAP_\tau$ and $PMAP_\tau$ coincide. Comparing Eq. (1.6) to Eq. (1.12), we see that the original and smooth duals are identical with the exception that max in the original is replaced with soft-max in the smoothed version. This is rather convenient as most of the structure that facilitated simple coordinate descent algorithms in the original dual can still be exploited.

Finally, we shall be interested in transformations between dual variables δ and primal variables μ (see Section 1.5). The following are the transformations obtained from the Lagrangian derivation (i.e., they can be used to switch from *optimal* dual variables to *optimal* primal variables).

$$\begin{aligned}
\mu_c(x_c;\delta) &\propto \exp\left(\tau\theta_c(x_c) - \tau\sum_{i:i\in c}\delta_{ci}(x_i)\right) \\
\mu_i(x_i;\delta) &\propto \exp\left(\tau\theta_i(x_i) + \tau\sum_{c:i\in c}\delta_{ci}(x_i)\right)
\end{aligned}$$

11. This results from the fact that the conjugate of the entropy function is the $\log\sum\exp$ function.

We denote the vector of all such marginals by $\mu(\delta)$. For the dual variables δ that minimize $F(\delta)$ it holds that $\mu(\delta)$ are feasible (i.e., $\mu(\delta) \in \mathcal{M}_L$). However, we will also consider $\mu(\delta)$ for non optimal δ, and show how to obtain primal feasible approximations from $\mu(\delta)$. These will be helpful in obtaining primal convergence rates.

It is easy to see that: $\left(\nabla F(\delta^t)\right)_{c,i,x_i} = \mu_i(x_i;\delta^t) - \mu_c(x_i;\delta^t)$, where (with some abuse of notation) we denote: $\mu_c(x_i) = \sum_{x_{c\setminus i}} \mu_c(x_{c\setminus i}, x_i)$. The elements of the gradient thus correspond to inconsistency between the marginals $\mu(\delta^t)$ (i.e., the degree to which they violate the constraints in Eq. (1.5)). We shall make repeated use of this fact to link primal and dual variables.

1.3 Coordinate Minimization Algorithms

In this section we propose several coordinate minimization procedures for solving $DMAP_\tau$ (Eq. (1.13)). We first set some notation to define block coordinate minimization algorithms. Denote the objective we want to minimize by $F(\delta)$ where δ corresponds to a set of N variables. Now define $\mathcal{S} = \{S_1, \ldots, S_M\}$ as a set of subsets, where each subset $S_i \subseteq \{1, \ldots, N\}$ describes a coordinate block. We will assume that $S_i \cap S_j = \emptyset$ for all i, j and that $\cup_i S_i = \{1, \ldots, N\}$.

Block coordinate minimization algorithms work as follows: at each iteration, first set $\delta^{t+1} = \delta^t$. Next choose a block S_i and set:

$$\delta^{t+1}_{S_i} = \operatorname*{argmin}_{\delta_{S_i}} F_i(\delta_{S_i}; \delta^t) \ , \tag{1.14}$$

where we use $F_i(\delta_{S_i}; \delta^t)$ to denote the function F restricted to the variables δ_{S_i} and where all other variables are set to their value in δ^t. In other words, at each iteration we fully optimize only over the variables δ_{S_i} while fixing all other variables. We assume that the minimization step in Eq. (1.14) can be solved in closed form, which is indeed the case for the updates we consider below.

Regarding the choice of an update schedule, several options are available:

- **Cyclic**: Decide on a fixed order (e.g., $S_1, \ldots, S_M$) and cycle through it.
- **Stochastic**: Draw an index i uniformly at random[12] at each iteration and use the block S_i.
- **Greedy**: Denote by $\nabla_{S_i} F(\delta^t)$ the gradient $\nabla F(\delta^t)$ evaluated at co-

12. Non uniform schedules are also possible. We consider the uniform for simplicity.

ordinates S_i only. The greedy scheme is to choose S_i that maximizes $\|\nabla_{S_i} F(\delta^t)\|_\infty$. In other words, choose the set of coordinates that correspond to maximum gradient of the function F. Intuitively this corresponds to choosing the block that promises the maximal (local) decrease in objective. Note that to find the best coordinate we presumably must process all sets S_i to find the best one. We will show later that this can be done rather efficiently in our case.

In our analysis, we shall focus on the Stochastic and Greedy cases, and analyze their rate of convergence. The cyclic case is typically hard to analyze, with results only under multiple conditions which do not hold here (e.g., see Saha and Tewari, 2013).

Another consideration when designing coordinate minimization algorithms is the choice of block size. One possible choice is all variables $\delta_{ci}(\cdot)$ (for a specific pair c, i). This is the block chosen in the max-sum-diffusion (MSD) algorithm (see Werner, 2007, 2009, for non-smooth and smooth MSD). A larger block that also facilitates closed form updates is the set of variables $\delta_{\cdot i}(\cdot)$. Namely, all messages into a variable i from factors c such that $i \in c$. We call this a *star* update. The update is used in Meshi et al. (2010) for the non-smoothed dual (but the possibility of applying it to the smoothed version is mentioned).

To derive the block updates, one needs to fix all variables except those in the block and then set the latter to minimize $F(\delta)$. Since $F(\delta)$ is differentiable this is pretty straightforward. The MSD update turns out to be:

$$\delta_{ci}^{t+1}(x_i) = \delta_{ci}^t(x_i) + \frac{1}{2\tau} \log \frac{\mu_c^t(x_i)}{\mu_i^t(x_i)}$$

for all x_i. Here we use $\mu_i^t(x_i), \mu_c^t(x_c)$ to denote the marginal obtained from Eq. (1.14) when using δ^t. Similarly, the star update is given by:

$$\delta_{ci}^{t+1}(x_i) = \delta_{ci}^t(x_i) + \frac{1}{\tau} \log \mu_c^t(x_i) - \frac{1}{N_i + 1} \cdot \frac{1}{\tau} \log \left(\mu_i^t(x_i) \cdot \prod_{c' : i \in c'} \mu_{c'}^t(x_i) \right)$$

for all $c : i \in c$ and all x_i, where $N_i = |\{c : i \in c\}|$.

It is interesting to consider the improvement in $F(\delta)$ as a result of an update. For the MSD update, it can be shown to be exactly:

$$F(\delta^t) - F(\delta^{t+1}) = -\frac{1}{\tau} \log \left(\sum_{x_i} \sqrt{\mu_i^t(x_i) \cdot \mu_c^t(x_i)} \right)^2 .$$

This is known as the *Bhattacharyya divergence measure* between the pair of distributions $\mu_i^t(x_i)$ and $\mu_c^t(x_i)$ (Bhattacharyya, 1946). Similarly, for the

star update the improvement in objective is exactly:

$$F(\delta^t) - F(\delta^{t+1}) = -\frac{1}{\tau}\log\left(\sum_{x_i}\left(\mu_i^t(x_i)\cdot\prod_{c:i\in c}\mu_c^t(x_i)\right)^{\frac{1}{N_i+1}}\right)^{N_i+1},$$

which is known as *Matusita's divergence measure* (Matusita, 1967), and is a generalization of the Bhattacharyya divergence to several distributions. Thus in both cases the improvement can be easily computed before actually applying the update and is directly related to how consistent the distributions $\mu_c^t(x_i), \mu_i^t(x_i)$ are. Recall that at the optimum they all agree since $\mu \in \mathcal{M}_L$, and thus the anticipated improvement is zero.

1.4 Dual Convergence Rate Analysis

We begin with the convergence rates of the dual F using greedy and random schemes described in Section 1.3. In Section 1.5 we subsequently show how to obtain a primal feasible solution and how the dual rates give rise to primal rates. Our analysis builds on the fact that we can lower bound the improvement at each step, as a function of some norm of the block gradient.

1.4.1 Greedy Block Minimization

Theorem 1.2. *Define B_1 to be a constant such that $\|\delta^t - \delta^*\|_1 \le B_1$ for all t. If there exists $k > 0$ so that* coordinate minimization *of each block S_i satisfies:*

$$F(\delta^t) - F(\delta^{t+1}) \ge \frac{1}{k}\|\nabla_{S_i} F(\delta^t)\|_\infty^2 \tag{1.15}$$

for all t, then for any $\epsilon > 0$ after $T = \frac{kB_1^2}{\epsilon}$ iterations of the greedy algorithm, $F(\delta^T) - F(\delta^*) \le \epsilon$.

Proof. Using Hölder's inequality we obtain the bound:

$$F(\delta^t) - F(\delta^*) \le \nabla F(\delta^t)^\top(\delta^t - \delta^*) \le \|\nabla F(\delta^t)\|_\infty \cdot \|\delta^t - \delta^*\|_1 .$$

This implies: $\|\nabla F(\delta^t)\|_\infty \ge \frac{1}{B_1}\big(F(\delta^t) - F(\delta^*)\big)$. Now, using the condition on the improvement and the greedy nature of the update, we obtain a bound

on the improvement:

$$\begin{aligned} F(\delta^t) - F(\delta^{t+1}) &\geq \frac{1}{k}\|\nabla_{S_i} F(\delta^t)\|_\infty^2 = \frac{1}{k}\|\nabla F(\delta^t)\|_\infty^2 \\ &\geq \frac{1}{kB_1^2}\left(F(\delta^t) - F(\delta^*)\right)^2 \\ &\geq \frac{1}{kB_1^2}\left(F(\delta^t) - F(\delta^*)\right)\left(F(\delta^{t+1}) - F(\delta^*)\right) \ . \end{aligned}$$

Hence,

$$\begin{aligned} \frac{1}{kB_1^2} &\leq \frac{F(\delta^t) - F(\delta^*) - \left(F(\delta^{t+1}) - F(\delta^*)\right)}{\left(F(\delta^t) - F(\delta^*)\right)\left(F(\delta^{t+1}) - F(\delta^*)\right)} \\ &= \frac{1}{F(\delta^{t+1}) - F(\delta^*)} - \frac{1}{F(\delta^t) - F(\delta^*)} \ . \end{aligned}$$

Summing over t we obtain:

$$\frac{T}{kB_1^2} \leq \frac{1}{F(\delta^T) - F(\delta^*)} - \frac{1}{F(\delta^0) - F(\delta^*)} \leq \frac{1}{F(\delta^T) - F(\delta^*)} \ ,$$

and the desired result follows. □

1.4.2 Stochastic Block Minimization

Theorem 1.3. *Define B_2 to be a constant such that $\|\delta^t - \delta^*\|_2 \leq B_2$ for all t. If there exists $k > 0$ so that coordinate minimization of each block S_i satisfies:*

$$F(\delta^t) - F(\delta^{t+1}) \geq \frac{1}{k}\|\nabla_{S_i} F(\delta^t)\|_2^2 \tag{1.16}$$

for all t, then for any $\epsilon > 0$ after $T = \frac{k|\mathcal{S}|B_2^2}{\epsilon}$ iterations of the stochastic algorithm we have that $\mathbb{E}[F(\delta^T)] - F(\delta^*) \leq \epsilon$, where the expectation is taken with respect to the randomization of blocks.

The proof is similar to Nesterov's analysis (see Theorem 1 in Nesterov (2010)). The proof in Nesterov (2010) relies on the improvement condition in Eq. (1.16) and not on the precise nature of the update. Note that since the cost of the update is roughly linear in the size of the block then this bound does not tell us which block size is better (the cost of an update times the number of blocks is roughly constant).

1.4.3 Analysis of $DMAP_\tau$ Block Minimization

We can now obtain rates for our coordinate minimization scheme for optimizing $DMAP_\tau$ by finding the k to be used in the conditions of Eqs. (1.15)

and (1.16). The results for the MSD and star updates are given below.

Proposition 1.4. *The MSD update satisfies the conditions in Eqs. (1.15) and (1.16) with* $k = 4\tau$.

Proposition 1.5. *The star update for variable* i *satisfies the conditions in Eqs. (1.15) and (1.16) with* $k = 4\tau N_i$.

This can be shown using Equation 2.4 in Nesterov (2004), which states that if $F_i(\delta_{S_i}; \delta)$ (see Eq. (1.14)) has Lipschitz constant L_i then Eq. (1.16) is satisfied with $k = 2L_i$. We can then use known bounds on the Lipschitz constant of the blocks (this can be calculated as in Savchynskyy et al. (2011)) to obtain the result.[13] To complete the analysis, it turns out that B_1 and B_2 can be bounded via a function of θ by bounding $\|\delta\|_1$ (see Appendix). We proceed to discuss the implications of these bounds.

1.4.4 Comparing the Different Schemes

The results we derived have several implications. First, we see that both stochastic and greedy schemes achieve a rate of $O(\frac{\tau}{\epsilon})$. This matches the known rates for regular (non-accelerated) gradient descent on functions with Lipschitz continuous gradient (e.g., see Nesterov (2004)), although in practice coordinate minimization is often much faster.

The main difference between the greedy and stochastic rates is that the factor $|\mathcal{S}|$ (the number of blocks) does not appear in the greedy rate, and does appear in the stochastic one. This can have a considerable effect since $|\mathcal{S}|$ is either the number of variables n (in the star update) or the sum of factor sizes $\sum_c |\{i : i \in c\}|$ (in MSD). Both can be significant (e.g., the number of edges in a pairwise MRF model). The greedy algorithm does pay a price for this advantage, since it has to find the optimal block to update at each iteration. However, for the problem we study here this can be done efficiently using a priority queue. To see this, consider the star update. A change in the variables $\delta_{\cdot i}(\cdot)$ will only affect the blocks that correspond to variables j that are in factors c such that $i \in c$. In many cases this is small (e.g., low degree pairwise MRFs) and thus we will only have to change the priority queue a small number of times, and this cost would be negligible when using a Fibonacci heap for example.[14] Indeed, our empirical results show that the greedy algorithm consistently outperforms the stochastic one

13. This can be also shown directly. For the MSD block see Kailath (1967), and for the star block we provide a direct proof in Meshi et al. (2012).
14. This was also used in the residual belief propagation approach (Elidan et al., 2006), which however is less theoretically justified than what we propose here.

(see Section 1.7).

As mentioned before, our analysis does not provide insight on which block size should be preferred, however in practice larger blocks usually perform better (see Section 1.7).

1.5 Primal Convergence

Thus far we have considered only dual variables. However, it is often important to recover the primal variables. We therefore focus on extracting primal feasible solutions from current δ, and characterize the degree of primal optimality and associated rates. The primal variables $\mu(\delta)$ (see Eq. (1.14)) need not be feasible in the sense that the consistency constraints in Eq. (1.5) are not necessarily satisfied. This is true also for other approaches to recovering primal variables from the dual, such as averaging subgradients when using subgradient descent (e.g., see Sontag et al., 2011).

We propose a simple two-step algorithm for transforming any dual variables δ into primal feasible variables $\tilde{\mu}(\delta) \in \mathcal{M}_L$. The resulting $\tilde{\mu}(\delta)$ will also be shown to converge to the optimal primal solution in Section 1.5.1. The procedure is described in Algorithm 1.1 below.

Algorithm 1.1 Mapping to a Feasible Primal Point

Step 1: Make marginals consistent.

For all i do: $\bar{\mu}_i(x_i) = \frac{1}{1+\sum_{c:i\in c}\frac{1}{|X_{c\setminus i}|}}\left(\mu_i(x_i) + \sum_{c:i\in c}\frac{1}{|X_{c\setminus i}|}\mu_c(x_i)\right)$

For all c do: $\bar{\mu}_c(x_c) = \mu_c(x_c) - \sum_{i:i\in c}\frac{1}{|X_{c\setminus i}|}\left(\mu_c(x_i) - \bar{\mu}_i(x_i)\right)$

Step 2: Make marginals non-negative.

$\lambda = 0$

for $c \in C, x_c$ **do**

  if $\bar{\mu}_c(x_c) < 0$ **then**

    $\lambda = \max\left\{\lambda, \frac{-\bar{\mu}_c(x_c)}{-\bar{\mu}_c(x_c)+\frac{1}{|X_c|}}\right\}$

  else if $\bar{\mu}_c(x_c) > 1$ **then**

    $\lambda = \max\left\{\lambda, \frac{\bar{\mu}_c(x_c)-1}{\bar{\mu}_c(x_c)-\frac{1}{|X_c|}}\right\}$

  end if

end for

for $\ell = 1, \ldots, n; c \in C$ **do**

  $\tilde{\mu}_\ell(x_\ell) = (1-\lambda)\bar{\mu}_\ell(x_\ell) + \lambda\frac{1}{|X_\ell|}$

end for

Importantly, all steps consist of cheap elementary local calculations in contrast to other methods previously proposed for this task (compare to Savchynskyy et al., 2011; Werner, 2011). The first step performs a Euclidian

projection of $\mu(\delta)$ to consistent marginals $\bar{\mu}$. Specifically, it solves:

$$\begin{array}{ll} \min_{\bar{\mu}} & \frac{1}{2}\|\mu(\delta) - \bar{\mu}\|^2 \\ \text{s.t.} & \bar{\mu}_c(x_i) = \bar{\mu}_i(x_i) \quad , \quad \forall\ c, i \in c, x_i \\ & \sum_i \bar{\mu}_i(x_i) = 1 \quad , \quad \forall i\ . \end{array} \tag{1.17}$$

Note that we did not include non-negativity constraints above, so the projection might result in negative $\bar{\mu}$. In the second step we "pull" $\bar{\mu}$ back into the feasible regime by taking a convex combination with the uniform distribution u (see Burshtein (2009) for a related approach). In particular, this step solves the simple problem of finding the smallest $\lambda \in [0, 1]$ such that $0 \leq \tilde{\mu} \leq 1$ (where $\tilde{\mu} = (1 - \lambda)\bar{\mu} + \lambda u$). Since this step interpolates between two distributions that satisfy consistency and normalization constraints, $\tilde{\mu}$ will be in the local polytope $\mathcal{M}_L$.

1.5.1 Primal Convergence Rate

Now that we have a procedure for obtaining a primal solution we analyze the corresponding convergence rate. First, we show that if we have δ for which $\|\nabla F(\delta)\|_\infty \leq \epsilon$ then $\tilde{\mu}(\delta)$ (after Algorithm 1) is an $O(\epsilon)$ primal optimal solution.

Theorem 1.6. *Denote by P^*_τ the optimum of the smoothed primal $PMAP_\tau$. For any set of dual variables δ, and any $\epsilon \in R(\tau)$ (see Appendix for definition of $R(\tau)$) it holds that if $\|\nabla F(\delta)\|_\infty \leq \epsilon$ then $P^*_\tau - P_\tau(\tilde{\mu}(\delta)) \leq C_0\epsilon$. The constant C_0 depends only on the parameters θ and is independent of τ.*

The proof is given in the Appendix. The key idea is to break $F(\delta) - P_\tau(\tilde{\mu}(\delta))$ into components, and show that each component is upper bounded by $O(\epsilon)$. The range $R(\tau)$ consists of $\epsilon \geq O(\frac{1}{\tau})$ and $\epsilon \leq O(e^{-\tau})$. As we show in the Appendix this range is large enough to guarantee any desired accuracy in the non-smoothed primal. We can now translate dual rates into primal rates. This can be done via the following well known lemma:

Lemma 1.7. *Any convex function F with Lipschitz continuous gradient and Lipschitz constant L satisfies $\|\nabla F(\delta)\|_2^2 \leq 2L\,(F(\delta) - F(\delta^*))$.*

These results lead to the following theorem.

Theorem 1.8. *Given any algorithm for optimizing $DMAP_\tau$ and $\epsilon \in R(\tau)$, if the algorithm is guaranteed to achieve $F(\delta^t) - F(\delta^*) \leq \epsilon$ after $O(g(\epsilon))$ iterations, then it is guaranteed to be ϵ primal optimal, i.e., $P^*_\tau - P_\tau(\tilde{\mu}(\delta^t)) \leq$*

ϵ after $O(g(\frac{\epsilon^2}{\tau}))$ iterations.[15]

Proof. Using $F(\delta^t) - F(\delta^*) \leq \epsilon$ and Lemma 1.7 we have that $\|\nabla F(\delta)\|_2^2 \leq 2L\epsilon$. Since the Lipschitz constant of $F(\delta)$ is $O(\tau)$, this implies $\|\nabla F(\delta)\|_2^2 \leq O(\tau\epsilon)$. We then use the fact that $\|\nabla F(\delta)\|_\infty^2 \leq \|\nabla F(\delta)\|_2^2$ to get $\|\nabla F(\delta)\|_\infty \leq O(\sqrt{\tau\epsilon})$. Finally, using Theorem 1.6 we obtain $P_\tau^* - P_\tau(\tilde{\mu}(\delta)) \leq O(\sqrt{\tau\epsilon})$, which completes the proof. □

The theorem lets us directly translate dual convergence rates into primal ones. Note that it applies to any algorithm for $DMAP_\tau$ (not only coordinate minimization), and the only property of the algorithm used in the proof is $F(\delta^t) \leq F(0)$ for all t. Put in the context of our previous results, any algorithm that achieves $F(\delta^t) - F(\delta^*) \leq \epsilon$ in $t = O(\tau/\epsilon)$ iterations is guaranteed to achieve $P_\tau^* - P_\tau(\tilde{\mu}(\delta^{t'})) \leq \epsilon$ in $t' = O(\tau^2/\epsilon^2)$ iterations.

1.6 The Augmented Dual LP Algorithm

An alternative approach to deal with the non-smoothness of the dual MAP-LP objective Eq. (1.6) is based on an augmented Lagrangian method known as the Alternating Direction Method of Multipliers (ADMM) (Glowinski and Marrocco, 1975; Gabay and Mercier, 1976; Boyd et al., 2011). Here we provide a short review of this approach and its application to MAP LP relaxations.

The ADMM framework is designed to handle convex optimization problems with the following constrained form:

$$\text{minimize } f(x) + g(z) \quad \text{s.t. } Ax = z \;, \tag{1.18}$$

where f and g are general convex functions.

The ADMM approach begins by adding the function $\frac{\rho}{2}\|Ax - z\|^2$ to the above objective, where $\rho > 0$ is a penalty parameter. This results in the optimization problem:

$$\text{minimize } f(x) + g(z) + \frac{\rho}{2}\|Ax - z\|^2 \quad \text{s.t. } Ax = z \;. \tag{1.19}$$

The augmenting quadratic term can be seen as smoothing the objective function. Clearly the above has the same optimum as Eq. (1.18) since when the constraints $Ax = z$ are satisfied, the added quadratic term equals zero.

15. We omit constants not depending on τ and ϵ.

The Lagrangian of the augmented problem Eq. (1.19) is given by:

$$\mathcal{L}_\rho(x, z, \nu) = f(x) + g(z) + \nu^\top (Ax - z) + \frac{\rho}{2} \|Ax - z\|^2 \ , \tag{1.20}$$

where ν is a vector of Lagrange multipliers. The solution to the problem of Eq. (1.19) is given by $\max_\nu \min_{x,z} \mathcal{L}_\rho(x, z, \nu)$. The ADMM method provides an elegant algorithm for finding this saddle point. The idea is to combine subgradient ascent over ν with coordinate descent over the x and z variables. The method applies the following iterations:

$$\begin{aligned} x^{t+1} &= \operatorname*{argmin}_x \mathcal{L}_\rho(x, z^t, \nu^t) \\ z^{t+1} &= \operatorname*{argmin}_z \mathcal{L}_\rho(x^{t+1}, z, \nu^t) \\ \nu^{t+1} &= \nu^t + \rho \left(Ax^{t+1} - z^{t+1}\right) \ . \end{aligned} \tag{1.21}$$

The algorithm consists of primal and dual updates, where the primal update is executed sequentially, minimizing first over x and then over z. This split retains the decomposition of the objective that has been lost due to the introduction of the quadratic term. Furthermore, it can be viewed as a coordinate descent algorithm with x and z blocks on $\mathcal{L}(x, z, \nu^t)$ for some fixed ν^t.

The ADMM algorithm is guaranteed to converge to the global optimum of Eq. (1.18) under rather mild conditions (Boyd et al., 2011). Moreover, it was recently shown that it has a convergence rate of $O(1/\epsilon)$ (He and Yuan, 2012; Wang and Banerjee, 2012), which is similar to accelerated gradient (Nesterov, 2005; Jojic et al., 2010), but does not require pre-smoothing of the objective.

There are various ways to apply ADMM to the dual LP Eq. (1.6). The challenge is to design the constraints in a way that facilitates efficient closed-form solutions for all updates. To this end, we duplicate the variables δ and denote the second copy by $\bar{\delta}$. We then introduce additional variables λ_c corresponding to the summation of δ's pertaining to factor c. To enforce overall agreement we introduce the constraints $\delta_{ci}(x_i) = \bar{\delta}_{ci}(x_i)$ for all $c, i : i \in c, x_i$, and $\lambda_c(x_c) = \sum_{i:i\in c} \bar{\delta}_{ci}(x_i)$ for all c, x_c.

Following the ADMM framework, we add quadratic terms and obtain the

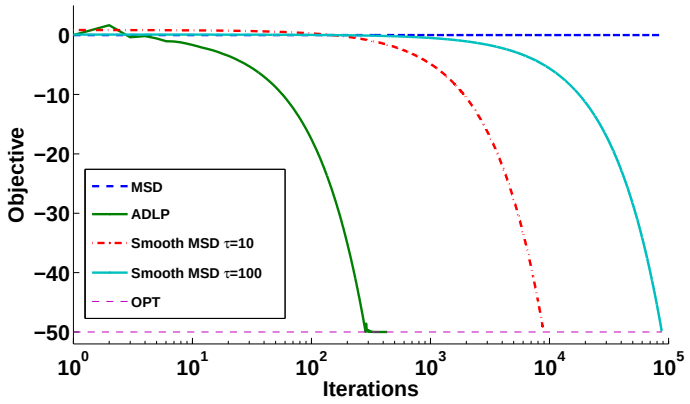


Figure 1.1: Comparison of smooth and non-smooth coordinate minimization algorithms on a toy MAP problem. The figure shows for each algorithm the dual objective as a function of the number of iterations. The optimal value of the non-smooth LP is marked with a thin dashed line.

augmented Lagrangian for the dual MAP-LP problem of Eq. (1.6):

$$\begin{aligned}
&\mathcal{L}_\rho(\delta,\lambda,\bar{\delta},\gamma,\mu) = \\
&\sum_i \max_{x_i}\left(\theta_i(x_i)+\sum_{c:i\in c}\delta_{ci}(x_i)\right)+\sum_c \max_{x_c}\left(\theta_c(x_c)-\lambda_c(x_c)\right) \\
&+\sum_c\sum_{i:i\in c}\sum_{x_i}\gamma_{ci}(x_i)\left(\delta_{ci}(x_i)-\bar{\delta}_{ci}(x_i)\right)+\frac{\rho}{2}\sum_c\sum_{i:i\in c}\sum_{x_i}\left(\delta_{ci}(x_i)-\bar{\delta}_{ci}(x_i)\right)^2 \\
&+\sum_c\sum_{x_c}\mu_c(x_c)\left(\lambda_c(x_c)-\sum_{i:i\in c}\bar{\delta}_{ci}(x_i)\right)+\frac{\rho}{2}\sum_c\sum_{x_c}\left(\lambda_c(x_c)-\sum_{i:i\in c}\bar{\delta}_{ci}(x_i)\right)^2 .
\end{aligned}$$

To see the relation of this formulation to Eq. (1.20), notice that the variables (δ,λ) correspond to x, the variables $\bar{\delta}$ correspond to z (with $g(z)=0$), and the multipliers (γ,μ) correspond to ν. The nice property of this decomposition is that all algorithmic steps in Eq. (1.21) can be done in simple closed form updates. These updates as well as a detailed derivation can be found in Meshi and Globerson (2011).

Finally, we note that ADMM can be also applied to the primal. e.g., see Martins et al. (2011) and Chapter by André Martins in this volume.

1.7 Experiments

In this section we evaluate the performance of coordinate minimization algorithms on toy and real-world MAP problems. We begin with a toy problem to demonstrate the effect of smoothing on the convergence of coordinate minimization. This toy problem was given in Kolmogorov (2006) (Appendix D therein) to illustrate that coordinate descent for the non-smooth dual LP can get stuck at non-optimal points. In Figure 1.1 we compare the convergence behavior of non-smooth MSD, smooth MSD, and ADLP.[16] We first see that non-smooth coordinate minimization (MSD) is caught in a suboptimal fixed point. In contrast, the smooth MSD algorithm is able to converge to the optimum of the smoothed dual objective Eq. (1.12). Figure 1.1 also shows the effect of the smoothing parameter τ. As τ increases the smoothed optimum gets closer to the LP optimum, but convergence time grows linearly with τ, as our analysis suggests. Finally, we observe that ADLP quickly converges to the optimum of the non-smooth LP.

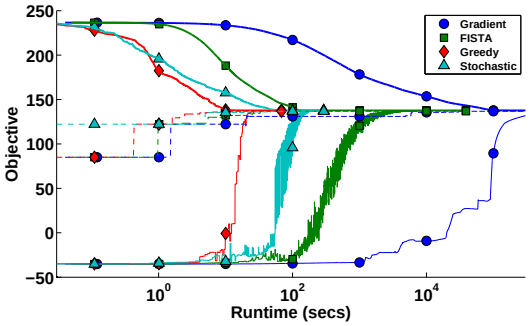
We next compare coordinate minimization algorithms to state-of-the-art baselines on a real-world MAP problem. Since the MSD block has similar or slightly inferior performance compared to the star block, we show here results only for the latter. We compare the running time of greedy coordinate minimization, stochastic coordinate minimization, full gradient descent, and FISTA – an accelerated gradient method (Beck and Teboulle, 2009). For completeness, we provide here the updates of both gradient-based algorithms:

Algorithm 1.2 Gradient descent

for $t = 1, \dots$ **do**
  $\delta^{t+1} = \delta^t - \frac{1}{L} \nabla F(\delta^t)$
end for

16. We run both MSD algorithms in stochastic schedule. Results are reported per iteration. Runtime is not identical to those, since the ADLP update is more costly.

(a)



(b) t_{alg}/t_{greedy}

Greedy	1
Stochastic	8.6 ± 0.6
FISTA	814.2 ± 38.1
Gradient	13849.8 ± 6086.5

Figure 1.2: Comparison of coordinate minimization, gradient descent, and the accelerated gradient algorithms on protein side-chain prediction task. Figure (a) shows a typical run of the algorithms. For each algorithm the dual objective of Eq. (1.12) is plotted as a function of execution time (upper solid line). The value (Eq. (1.8)) of the feasible primal solution of Algorithm 1.1 is also shown (lower solid line), as well as the objective (Eq. (1.3)) of the best decoded integer solution (dashed line; those are decoded directly from the dual variables δ). Table (b) shows the ratio of runtime of each algorithm w.r.t. the greedy algorithm. The mean ratio over the proteins in the dataset is shown followed by standard error.

Algorithm 1.3 FISTA

1: $\bar{\delta}^1 = \delta^0, \quad \alpha^1 = 1$
2: **for** $t = 1, \ldots$ **do**
3: $\quad \delta^t = \bar{\delta}^t - \frac{1}{L}\nabla F(\bar{\delta}^t)$
4: $\quad \alpha^{t+1} = \frac{1+\sqrt{1+4(\alpha^t)^2}}{2}$
5: $\quad \bar{\delta}^{t+1} = \delta^t + \left(\frac{\alpha^t - 1}{\alpha^{t+1}}\right)\left(\delta^t - \delta^{t-1}\right)$
6: **end for**

Gradient descent is known to converge in $O\left(\frac{1}{\epsilon}\right)$ iterations while FISTA converges in $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ iterations (Beck and Teboulle, 2009). We compare the performance of the algorithms on protein side-chain prediction problems from the dataset of Yanover et al. (2006). These problems involve finding the 3D configuration of rotamers given the backbone structure of a protein. The problems are modeled by singleton and pairwise factors and can be posed as finding a MAP assignment for the given model.

Figure 1.2(a) shows the objective value for each algorithm over time. We first notice that the greedy algorithm converges faster than the stochastic one. This is in agreement with our theoretical analysis. Second, we observe that the coordinate minimization algorithms are competitive with the accelerated gradient method FISTA and are much faster than the gradient

method. Third, as Theorem 1.8 predicts, primal convergence is slower than dual convergence (notice the logarithmic timescale). Finally, we can see that better convergence of the dual objective corresponds to better convergence of the primal objective, in both fractional and integral domains. In our experiments the quality of the decoded integral solution (dashed lines) significantly exceeds that of the fractional solution. Although sometimes a fractional solution can be useful in itself, this suggests that if only an integral solution is sought then it could be enough to decode directly from the dual variables.

The table in Figure 1.2(b) shows overall statistics for the proteins in the dataset. Here we run each algorithm until the duality gap drops below a fixed desired precision ($\epsilon = 0.1$) and compare the total runtime. The table presents the ratio of runtime of each algorithm w.r.t. the greedy algorithm (t_{alg}/t_{greedy}). These results are consistent with the example in Figure 1.2(a).

1.8 Discussion

The chapter provided a tutorial introduction to the smoothing approach to MAP LP relaxations. It was shown that coordinate descent on the smoothed dual results in simple updates, and that the rate of convergence can be analyzed for different coordinate update schedules.

We also showed how such dual iterates can be turned into primal feasible iterates and analyzed the rate with which these primal iterates converge to the primal optimum. The primal mapping is of considerable practical value, as it allows us to monitor the distance between the upper (dual) and lower (primal) bounds on the optimum and use this as a stopping criterion. Note that this cannot be done without a primal feasible solution. An alternative commonly used progress criterion is to decode an integral solution from the dual variables, and see if its value is close to the dual upper bound. However, this will only work if $PMAP$ has an integral solution and we have managed to decode it.

The overall rates we obtain are of the order $O(\frac{\tau}{\epsilon})$ for the $DMAP_\tau$ problem. If one requires an ϵ accurate solution for $PMAP$, then τ needs to be set to $O(\frac{1}{\epsilon})$ (see Eq. (1.9)) and the overall rate is $O(\frac{1}{\epsilon^2})$ in the dual. As noted in Jojic et al. (2010); Savchynskyy et al. (2011), a faster rate of $O(\frac{1}{\epsilon})$ may be obtained using accelerated methods such as Nesterov's (Nesterov, 2005) or FISTA (Beck and Teboulle, 2009). However, these also have an extra factor of N which does not appear in the greedy rate. This could partially explain the excellent performance of the greedy scheme when compared to FISTA (see Section 1.7).

As mentioned earlier, there are various ways of choosing the block of

coordinates to update. Here we focused mainly on the *star* update where a variable i is chosen and $\delta_{ci}(x_i)$ are updated for all values of c, x_i. An alternative choice is to choose c, i and update $\delta_{ci}(x_i)$ for all x_i values. This corresponds to the MSD update of Werner (2007). While we have provided some of the results for the MSD case (e.g., see Proposition 1.4), we did not analyze the overall runtime expected for the two variants (e.g., for the greedy scheme, different queue maintenance costs will be incurred). Empirically, we observed that the star update is typically faster, as may intuitively be expected due to more coordinates being updated. Understanding the effect of block length and the resulting tradeoffs is an interesting problem for further study.

Another block update strategy is the so-called MPLP update, where for a given c, the variables $\delta_{ci}(x_i)$ are updated for all $i \in c, x_i$. Interestingly, we were unable to obtain these in closed form for the particular entropy smoothing we use here (i.e., for the dual in Eq. (1.12)). It would be interesting to seek primal regularization schemes where such coordinate blocks have closed form updates.

The main goal of smoothing was to obtain a dual that is differentiable, and where coordinate descent globally converges. The way this is achieved here is by introducing entropy regularization into the primal. It would be interesting to study other forms of primal regularization and the resulting smooth duals. For example, one may consider ℓ_2 regularization on μ. It is not clear however, that closed form updates are available in this case for dual coordinate minimization. An additional regularization form to consider is ℓ_2 regularization on δ in the dual.

Both our empirical and theoretical results highlight the advantage of greedy update schedules. The advantage comes from the fact that the choice of block to update is quite efficient since its cost is of the order of the other computations required by the algorithm. This can be viewed as a theoretical reinforcement of selective scheduling algorithms such as Residual Belief Propagation (Elidan et al., 2006).

The convergence rates we provided were for obtaining an accuracy ϵ with respect to the primal or dual objective optima. However, in analyzing combinatorial optimization problems with polynomial time algorithms, one typically obtains the number of iterations required to find the optimal (discrete) solution, without reference to accuracy. To obtain such runtimes in our case, we can focus on problems where the LP is integral (e.g., mincut; See proof in Taskar et al., 2006). If the LP solution is close enough to optimal, this can be shown to imply that it can be rounded to the optimal integral

assignment.[17] Such an analysis was performed in Ravikumar et al. (2010) and can be done in our case as well.

Finally, our analysis relates to sequential updates of coordinates. Thus, it is not immediately applicable to distributed asynchronous implementations. It would be very interesting to extend the results to the distributed setting (e.g., Bradley et al., 2011).

Acknowledgments:

This work was supported by BSF grant 2008303. Ofer Meshi is a recipient of the Google Europe Fellowship in Machine Learning, and this research was supported in part by this Google Fellowship.

Appendix: Primal Convergence Rate

In this section we prove Theorem 1.6.

Proof. $\|\nabla F(\delta)\|_\infty \leq \epsilon$ guarantees that $\mu = \mu(\delta)$ are ϵ-consistent in the sense that $|\mu_i(x_i) - \mu_c(x_i)| \leq \epsilon$ for all $c, i \in c$ and x_i. Algorithm 1.1 maps any such ϵ-consistent μ to locally consistent marginals $\tilde{\mu}$ such that

$$|\mu_i(x_i) - \tilde{\mu}_i(x_i)| \leq 3\epsilon N_{\max}, \quad |\mu_c(x_c) - \tilde{\mu}_c(x_c)| \leq 2\epsilon N_{\max}^2, \tag{1.22}$$

for all i, x_i, c, and x_c, where $N_{\max} = \max\{\max_i N_i, \max_c N_c\}$. In other words, $\|\mu - \tilde{\mu}\|_\infty \leq K\epsilon$. This can be easily derived from the update in Algorithm 1.1 and the fact that $|\mu_i(x_i) - \mu_c(x_i)| \leq \epsilon$.

Next, it can be shown that $F(\delta) = P_\tau(\mu(\delta))$. And it follows that $P_\tau^* \leq F(\delta) \leq P_\tau(\mu)$, where the first inequality follows from weak duality.

For clarity, we define

$$\mu \cdot \theta = \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_c \sum_{x_c} \mu_c(x_c)\theta_c(x_c) \tag{1.23}$$

$$H(\mu) = \sum_i H(\mu_i(\cdot)) + \sum_c H(\mu_c(\cdot)) \tag{1.24}$$

17. Assuming a unique integral optimum.

Thus we have:

$$\begin{aligned} P_\tau^* \leq P_\tau(\mu) &= \mu \cdot \theta + \frac{1}{\tau} H(\mu) \\ &= (\tilde{\mu} + \mu - \tilde{\mu}) \cdot \theta + \frac{1}{\tau} H(\tilde{\mu}) + \frac{1}{\tau}(H(\mu) - H(\tilde{\mu})) \\ &\leq P_\tau(\tilde{\mu}) + \|\mu - \tilde{\mu}\|_\infty \|\theta\|_1 + \frac{1}{\tau}(H(\mu) - H(\tilde{\mu})) \\ &\leq P_\tau(\tilde{\mu}) + K\epsilon\|\theta\|_1 + \frac{1}{\tau}(H(\mu) - H(\tilde{\mu})) \end{aligned} \tag{1.25}$$

Where we have used Hölder's inequality for the first inequality and Eq. (1.22) for the second inequality.

It remains to bound $\frac{1}{\tau}(H(\mu) - H(\tilde{\mu}))$ by a linear function of ϵ. We note that it is impossible to achieve such a bound in general (e.g., see Berend and Kontorovich (2012)). However, since the entropy is bounded the difference is also bounded. Now, if we also restrict ϵ to be large enough $\epsilon \geq \frac{1}{\tau}$, then we obtain the bound:

$$\frac{1}{\tau}(H(\mu) - H(\tilde{\mu})) \leq \frac{1}{\tau} H_{\max} \leq \epsilon H_{\max} \tag{1.26}$$

We thus obtain that Eq. (1.25) is of the form $P_\tau(\tilde{\mu}) + O(\epsilon)$ and the result follows.

For the high-accuracy regime (small ϵ) we provide a similar bound for the case $\epsilon \leq O(e^{-\tau})$. Let $v = \mu - \tilde{\mu}$, so we have:

$$\begin{aligned} H(\mu) - H(\tilde{\mu}) &= H(\tilde{\mu} + v) - H(\tilde{\mu}) \\ &\leq H(\tilde{\mu}) + \nabla H(\tilde{\mu})^\top v - H(\tilde{\mu}) \\ &= -\sum_i \sum_{x_i} v_i(x_i) \log \tilde{\mu}_i(x_i) - \sum_c \sum_{x_c} v_c(x_c) \log \tilde{\mu}_c(x_c) \end{aligned}$$

where the inequality follows from the concavity of entropy, and the second equality is true because $\sum_{x_i} v_i(x_i) = 0$ and similarly for $v_c(x_c)$. Now, from the definition of $\mu_i(x_i; \delta)$ we obtain the following bound:

$$\mu_i(x_i; \delta) = \frac{1}{Z_i} e^{\tau(\theta_i(x_i) + \sum_{c: i \in c} \delta_{ci}(x_i))} \geq \frac{1}{|X_i|} e^{-2\tau(\|\theta_i\|_\infty + \|\delta_i\|_1)}$$

We will show below (Lemma 1.9) that $\|\delta_i\|_1$ remains bounded by a constant A independent of τ. Thus we can write:

$$\mu_i(x_i; \delta) \geq \frac{1}{|X_{\max}|} e^{-2\tau(\|\theta_i\|_\infty + A)}$$

where $|X_{\max}| = \max\{\max_i |X_i|, \max_c |X_c|\}$. We define $\gamma_0 = \frac{1}{(2|X_{\max}|)^\tau} e^{-2\tau(\|\theta_i\|_\infty + A)}$, and thus for any $\tau \geq 1$ we have that $\mu_i(x_i; \delta)$ is bounded away from zero by $2^\tau \gamma_0$. Since we assume that $\epsilon \leq \gamma_0$, we can bound $\tilde{\mu}$ from below by γ_0. As a

result, since $\|v_i\|_\infty \leq K\epsilon$,

$$\begin{aligned} -\frac{1}{\tau}\sum_i \sum_{x_i} v_i(x_i)\log\tilde{\mu}_i(x_i) &\leq -\frac{1}{\tau}(\log\gamma_0)|X_i|K\epsilon \\ &= (2(\|\theta_i\|_\infty + A) + \log(2|X_{\max}|))|X_i|K\epsilon \end{aligned}$$

and similarly for the other entropy terms.

Again, we obtain that Eq. (1.25) is of the form $P_\tau(\tilde{\mu}) + O(\epsilon)$ and the result holds.

In conclusion, we have shown that if $\|\nabla F(\delta)\|_\infty \leq \epsilon$, then for large values $\epsilon \geq \frac{1}{\tau}$ and small values $\epsilon \leq \frac{1}{(2|X_{\max}|)^\tau} e^{-2\tau(\|\theta_i\|_\infty + A)}$ we have that: $P^*_\tau - P_\tau(\tilde{\mu}) \leq O(\epsilon)$. Our analysis does not cover values in the middle range, but we next argue that the covered range is useful. □

The allowed range of ϵ (namely $\epsilon \in R(\tau)$) seems like a restriction. However, as we argue next taking $\epsilon \geq \frac{1}{\tau}$ (i.e., $\epsilon \in R(\tau)$) is all we need in order to obtain a desired accuracy in the non-smoothed primal.

Suppose one wants to solve the original problem $PMAP$ to within accuracy ϵ'. There are two sources of inaccuracy, namely the smoothing and suboptimality. To ensure the desired accuracy, we require that $P^*_\tau - P^* \leq \alpha\epsilon'$ and likewise $P_\tau(\tilde{\mu}) - P^*_\tau \leq (1-\alpha)\epsilon'$. In other words, we allow $\alpha\epsilon'$ suboptimality due to smoothing and $(1-\alpha)\epsilon'$ due to suboptimality.

For the first condition, it is enough to set the smoothing constant as: $\tau \geq \frac{H_{\max}}{\alpha\epsilon'}$. The second condition will be satisfied as long as we use an ϵ such that: $\epsilon \leq \frac{(1-\alpha)\epsilon'}{(K\|\theta\|_1 + H_{\max})}$ (see Eq. (1.25) and Eq. (1.26)). If we choose $\alpha = \frac{H_{\max}}{K\|\theta\|_1 + 2H_{\max}}$ we obtain that this ϵ satisfies $\epsilon \geq \frac{1}{\tau}$ and therefore $\epsilon \in R(\tau)$.

Lemma 1.9. *Assume δ is a set of dual variables satisfying $F(\delta) \leq F(0)$ where $F(0)$ is the dual value corresponding to $\delta = 0$. We can require $\sum_{c:i\in c}\delta_{ci}(x_i) = 0$ since $F(\delta)$ is invariant to constant shifts. Then it holds that:*

$$\sum_{c,i,x_i} |\delta_{ci}(x_i)| = \|\delta\|_1 \leq A \tag{1.27}$$

where

$$A = 2\max_i |X_i| \left(F(0) + \sum_i \max_{x_i} |\theta_i(x_i)| + \sum_c \max_{x_c} |\theta_c(x_c)| \right) \tag{1.28}$$

Proof. To show this, we bound

$$\begin{aligned} &\max_{\delta} \sum_{c,i,x_i} r_{ci}(x_i)\delta_{ci}(x_i) \\ &\text{s.t. } F(\delta) \leq F(0) \\ &\quad \sum_{c:i\in c} \delta_{ci}(x_i) = 0 \end{aligned} \tag{1.29}$$

For any $r_{ci}(x_i) \in [-1, 1]$. The dual problem turns out to be:

$$\begin{aligned} \min_{\mu,\gamma,\alpha} \quad & \alpha(F(0) - \sum_{c,x_c} \mu_c(x_c)\theta_c(x_c) - \sum_{i,x_i} \mu_i(x_i)\theta_i(x_i) - \sum_i H(\mu_i(x_i)) - \sum_c H(\mu_c(x_c)) \\ \text{s.t.} \quad & \mu_i(x_i) - \mu_c(x_i) = \tfrac{r_{ci}(x_i)-\gamma_{ci}}{\alpha} \\ & \mu_i(x_i) \geq 0, \mu_c(x_c) \geq 0 \\ & \textstyle\sum_{x_i} \mu_i(x_i) = 1, \sum_{x_c} \mu_c(x_c) = 1 \\ & \alpha \geq 0 \end{aligned} \tag{1.30}$$

We will next upper bound this minimum with a constant independent of r and thus obtain an upper bound that holds for all r. To do this, we will present a feasible assignment to the variables α, μ, γ above and use the value they attain. First, we set $\alpha = \hat{\alpha} = 2\max_i |X_i|$. Next, we note that for this $\hat{\alpha}$, the objective of Eq. (1.30) is upper bounded by A (as defined in Eq. (1.28)). Thus we only need to show that $\hat{\alpha} = 2\max_i |X_i|$ is indeed a feasible value, and this will be done by showing feasible values for the other variables denoted by $\hat{\mu}, \hat{\gamma}$. First, we set:

$$\hat{\mu}_i(x_i) \quad = \quad \frac{1}{|X_i|}$$

and:

$$\hat{\gamma}_{ci} = \frac{1}{|X_i|} \sum_{x_i} r_{ci}(x_i) \tag{1.31}$$

Next, we define $\nu_{ci}(x_i)$ (for all c, i, x_i) as follows:

$$\nu_{ci}(x_i) = \hat{\mu}_i(x_i) - \frac{r_{ci}(x_i) - \hat{\gamma}_{ci}}{\hat{\alpha}} \tag{1.32}$$

It can easily be shown that $\nu_{ci}(x_i)$ is a valid distribution over x_i (i.e., non negative and sums to one). Thus we can define:

$$\hat{\mu}_c(x_c) = \prod_{i\in c} \nu_{ci}(x_i) \tag{1.33}$$

Since $\hat{\mu}_c(x_c)$ is a product of distributions over the variables in c, it is also a valid distribution. Thus it follows that all constraints in Eq. (1.30) are satisfied by $\hat{\alpha}, \hat{\gamma}, \hat{\mu}$, and the desired bound holds.

□

1.10 References

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, Mar. 2009.

D. Berend and A. Kontorovich. A reverse Pinsker inequality. *CoRR*, abs/1206.6544, 2012.

A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhy: The Indian Journal of Statistics (1933-1960)*, 7(4):pp. 401–406, 1946.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

S. Boyd, N. Parikh, and E. Chu. *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Now Publishers, 2011.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 1999.

J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *International Conference on Machine Learning (ICML 2011)*, Bellevue, Washington, June 2011.

D. Burshtein. Iterative approximate linear programming decoding of LDPC codes with linear complexity. *IEEE Transactions on Information Theory*, 55(11):4835–4859, 2009.

K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Trans. Algorithms*, 6(4):63:1–63:30, Sept. 2010.

G. Elidan, I. Mcgraw, and D. Koller. Residual belief propagation: informed scheduling for asynchronous message passing. In *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, 2006.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Computers and Mathematics with Applications*, 2:17–40, 1976.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *NIPS 20*. MIT Press, 2008.

R. Glowinski and A. Marrocco. Sur lapproximation, par elements finis dordre un, et la resolution, par penalisation-dualité, dune classe de problems de dirichlet non lineares. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, 9:4176, 1975.

T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010.

B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50(2):700–709, Apr. 2012.

V. Jojic, S. Gould, and D. Koller. Fast and smooth: Accelerated dual decomposition for MAP inference. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52 –60, February 1967.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10): 1568–1583, 2006.

V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33:531–552, March 2011.

A. L. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing. An augmented Lagrangian approach to constrained MAP inference. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 169–176, 2011.

K. Matusita. On the notion of affinity of several distributions and some of its applications. *Annals of the Institute of Statistical Mathematics*, 19:181–192, 1967.

O. Meshi and A. Globerson. An alternating direction method for dual MAP LP relaxation. In *ECML PKDD*, pages 470–483. Springer-Verlag, 2011.

O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson. Learning efficiently with approximate inference via dual losses. In *Proceedings of the 27th International Conference on Machine Learning*, pages 783–790, New York, NY, USA, 2010. ACM.

O. Meshi, T. Jaakkola, and A. Globerson. Convergence rate analysis of MAP coordinate minimization algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3023–3031. 2012.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Kluwer Academic Publishers, 2004.

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

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. Core discussion papers, Universit Catholique de Louvain, 2010.

P. Ravikumar, A. Agarwal, and M. J. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research*, 11:1043–1080, Mar. 2010.

A. Saha and A. Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

B. Savchynskyy, S. Schmidt, J. Kappes, and C. Schnörr. A study of Nesterov's scheme for lagrangian decomposition and MAP labeling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, July 2011.

D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In *Optimization for Machine Learning*, pages 219–254. MIT Press, 2011.

B. Taskar, S. Lacoste-Julien, and M. Jordan. Structured prediction, dual extragradient and Bregman projections. *Journal of Machine Learning Research*, pages

1627–1653, 2006.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization 1. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

M. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA, 2008.

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

H. Wang and A. Banerjee. Online alternating direction method. In *ICML*, 2012.

Y. Weiss, C. Yanover, and T. Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 416–425, Arlington, Virginia, 2007. AUAI Press.

T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, 2007.

T. Werner. Revisiting the decomposition approach to inference in exponential families and graphical models. Technical Report CTU-CMP-2009-06, Czech Technical University, 2009.

T. Werner. How to compute primal solution from dual one in MAP inference in MRF? In *Control Systems and Computers (special issue on Optimal Labeling Problems in Structual Pattern Recognition)*, 2011.

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