Solving Markov Random Fields with Spectral Relaxation

Timothee Cour
Computer and Information Science Dept.
University of Pennsylvania
Philadelphia, PA 19104

Jianbo Shi
Computer and Information Science Dept.
University of Pennsylvania
Philadelphia, PA 19104

Abstract

Markov Random Fields (MRFs) are used in a large array of computer vision applications. Finding the Maximum Aposteriori (MAP) solution of an MRF is in general intractable, and one has to resort to approximate solutions, such as Belief Propagation, Graph Cuts, or more recently, approaches based on quadratic programming. We propose a novel type of approximation, Spectral relaxation to Quadratic Programming (SQP). We show our method offers tighter bounds than recently published work, while at the same time being computationally efficient. We compare our method to other algorithms on random MRFs in various settings.

1 Introduction

A number of problems in Computer Vision and Machine Learning can be formulated in a probabilistic setting using Markov Random Fields (MRF). Classical examples include stereo vision, image restoration, image labeling and graph matching. In each case, a set of interdependant variables can be assigned a range of labels, with a probability attached to each joint assignment. Inference in such a graphical model consists in finding the configuration with maximum a posteriori probability (MAP). In general, the inference problem is intractable, but there are interesting cases where it can be solved in polynomial time, such as tree-structured MRFs, MRFs with convex priors[1], or binary MRFs with submodular clique potentials.

MRFs have been studied extensively since the 1970's, and a lot of work has been focused on developping approximation algorithms for the MAP problem. Bayesian methods such as Belief Propagation (BP)[2, 3], Generalized BP and Tree Reweighted BP are optimal in trees as well as certain graphs with cycles. In the general case, when the max-product version of BP converges, the assignment is guaranteed to be locally optimal in a large neighborhood[3]. However, there is no general convergence guarantee and BP may fail to converge even in simple graphs. Energy Minimization methods such as Graph Cuts[4, 5] have been successfully applied to early vision applications, often on planar graphs with nearest neighbor connectivity. For binary MRFs with submodular clique potentials, Graph Cuts are provably optimal. For multiple label MRFs, [4] introduces $\alpha - \beta$ swaps and α exansion moves that find solutions which are locally optimal with respect to large moves, but with some restrictions on the clique potentials.

In this paper we present a new algorithm, Spectral relaxation to Quadratic Programming (SQP) to solve the MAP problem approximately. Unlike Graph Cuts and BP, there are *no restrictions on the clique potentials* and the graph can have *arbitrary topology*. Our algorithm is quite simple and very efficient, with complexity $O(\#edges \#labels^2)$, *linear in the description length* of the clique potentials. We show our method *improves optimality bounds* over recently published literature, and confirm this with experiments. As a by product, we give a complete treatment of a new class of problems, maximization of rayleigh quotients under *affine* constraints, generalizing the *linear* constraint case, and we show furthermore that the same problem under *inequality constraints* is NP-hard.

Related to our work are relaxation methods that attempt to solve the MRF in the continuous domain, such as Relaxation Labeling[6], Deterministic annealing and LP relaxation[7, 8]. Our work drives its main inspiration from CQP[9] and L2QP[10, 11], and we will go over them in more details in section 3. All those methods typically start by reformulating the MAP problem as an Integer Quadratic Program (IQP) and then relax the integral constraint into a Quadratic Program (QP). *We will show that in fact IQP is equiv-*

alent to QP, under more general conditions than was recently established in[9, 11]. The resulting QP is usually non-convex and NP-hard, and further approximations are needed, which is the goal of all those methods.

The paper is organized as follows. Section 2 formulates the MAP-MRF problem and shows how to reduce it to a Quadratic Program (QP). Section 3 summarizes the approximation algorithms that try to solve the QP. Section 4 introduces our new Spectral relaxation to Quadratic Programming (SQP) algorithm. We analyse its optimality guarantees and properties in sections 5 and 6 and report experiments in section 7.

2 Problem formulation and preliminaries

General MRF formulation We review here the general MRF formulation with unary and binary clique potentials. Let G be an undirected graph with n vertices or sites, and edge set E. We attach to each vertex i a random variable $X_i \in \{1, ..., k\}$[1], designing the state of that site. A set of binary and unary potential functions Ψ_{ij} and Φ_i determine compatibility of assignments of neighboring or individual vertices. The joint distribution represented by the MRF is:

$$P(X) = \frac{1}{Z} \prod_{ij \in E} \Psi_{ij}(X_i, X_j) \prod_i \Phi_i(X_i), \qquad (1)$$

where Z is a normalization constant. The Maximum Aposteriori (MAP) inference problem is to maximize $P(X)$ over all possible joint assignments $X \in \{1, ..., k\}^n$.

2.1 IQP formulation

We show here how to rewrite the MAP problem as an Integer Quadratic Programming (IQP), which is easier for us to deal with. Let $x_{ia} \in \{0, 1\}$ be a binary random variable with $x_{ia} = 1$ iff $X_i = a$. We concatenate each x_{ia} as a vector $x = (x_{ia})$. Since each site can take a single state, we have the constraint $\sum_a x_{ia} = 1$, which we can rewrite as a linear constraint $Cx = 1$ for a certain matrix C. Next, we introduce the $nk \times nk$ matrix W as $W_{iajb} = \log \Psi_{ij}(a, b)$ (if $ij \notin E, W_{iajb} = 0$), and the $nk \times 1$ vector V as $V_{ia} = \log \Phi_i(a)$. WLOG, we assume W symmetric. With these notations, $\log P(X) = \sum_{ij \in E} W_{iajb} x_{ia} x_{jb} + \sum_i V_{ia} x_{ia} + constant$ and the MAP problem becomes:

$$\max \epsilon(x) = x^\mathsf{T} W x + V^\mathsf{T} x, \quad \text{s.t.} \quad Cx = 1, x \in \{0, 1\}^{nk} \qquad (2)$$

[1]It is straightforward to extend the following to the case where each site can have a variable number of labels, i.e. $X_i \in \{1, ..., k_i\}$

In general this IQP is NP-hard, and approximate solutions are needed. An interesting yet counterintuitive fact is that *we can remove the discrete constraint without changing the problem*, as we shall see in the next section. First, let us introduce some notations.

Definitions Let $\Omega_a = \{x \in \mathbb{R}^{nk} : Cx = 1\}$, $\Omega_s = \{x \in \Omega_a : x \geq 0\}$, $\Omega_d = \Omega_a \cap \{0, 1\}^{nk}$. Note, Ω_d denotes the feasible (*discrete*) points of the IQP, and Ω_a, Ω_s are relaxations of Ω_d (using resp. *affine* subspace and the standard *simplex*).

2.2 QP relaxation

The QP relaxation relaxes the set Ω_d to Ω_s in (2):

$$\max \quad \epsilon(x), \quad \text{s.t.} \quad Cx = 1, 0 \leq x \leq 1 \qquad (3)$$

We extend in the following proposition some recent results from [11, 9], which only considered the case $W_{iaib} = 0, \forall i, a, b$. Such terms can affect the solution of the QP (3), which motivates this new result.

Proposition 2.1 (QP is equivalent to IQP) *Suppose $W_{iaia} \geq 2W_{iaib}\ \forall a \neq b$, all other entries in W being unconstrained. Then from any $x \in \Omega_s$, we can construct efficiently $x_d \in \Omega_d$ such that $\epsilon(x_d) \geq \epsilon(x)$. As a corollary, $\max_{x \in \Omega_s} \epsilon(x) = \max_{x_d \in \Omega_d} \epsilon(x_d)$ and (3) is equivalent to (2).*

In the rest of the paper, *we assume WLOG that the condition $W_{iaia} \geq 2W_{iaib}\ \forall a \neq b$ is always met*, since it is easy to see that terms of the form W_{iaib} with $a \neq b$ have no influence in the IQP (2).

Proof of proposition 2.1 The proof uses a construction similar to ICM (Iterative Conditional Modes)[12], but requires special treatment for terms of the form W_{iaib}. Let $y^0 = x$, and, for $t = 1..n$ let $y^t_{ia} = y^{t-1}_{ia}$ except for $i = t$: let $v_a = 2 \sum_{(j,b) \neq (t,a)} W_{tajb}\ y^{t-1}_{jb} + W_{tata}\ y^{t-1}_{ta} + V_{ta}$ and $c = \arg\max_a v_a$. We take $y^t_{tc} = 1$ and $y^t_{ta} = 0$ for $a \neq c$. One can verify that $\epsilon(y^t) \geq \epsilon(y^{t-1})$. The only non-trivial thing to see is that $2 \sum_{b \neq c} W_{tctb}\ y^{t-1}_{tb} + W_{tctc}\ y^{t-1}_{tc} \leq W_{tctc}$ because of the hypothesis and the fact that all $y^t \in \Omega_s$. Finally, we take $x_d := y^{(n)} \in \Omega_d$. The corollary comes from the fact that $\max_{\Omega_s} \epsilon \geq \max_{\Omega_d} \epsilon$ □

In general, solving the QP is still NP-hard. We briefly review a few recent approximation algorithms, before presenting our own contribution to the problem.

3 Previous Work to approximate the Quadratic Program

We present here recent attempts to solve the QP (3) that are most relevant to our work.

Linear relaxations: LP, SDP, SOCP The QP can be rewritten as a (linear) matrix inner product: $x^\mathsf{T} W x + V^\mathsf{T} x = \langle X, W_{eq} \rangle$ where $X = [x;1]^\mathsf{T}[x;1]$ is constrained to be rank 1 (as well as additional affine constraints). The **LP** relaxation [7, 8] approximates the non-convex rank 1 constraint by affine *local consistency* constraints. The authors show its relation to tree-reweighted belief propagation, and state conditions for optimality. The **SDP** relaxation [13] attempts to find a tighter relaxation by approximating $X = [x;1]^\mathsf{T}[x;1]$ to $X \succeq [x;1]^\mathsf{T}[x;1]$, but suffers from expensive SDP solvers. The **SOCP** relaxation [14] proposes a more efficient method than SDP by further relaxing $X \succeq [x;1]^\mathsf{T}[x;1]$ to $\langle X, S \rangle \succeq [x;1]^\mathsf{T} S [x;1]$ for a suitable choice of symmetric matrices $S \in \mathbb{S}$. Note, all these methods suffer from the fact that the *number of variables is squared* (although SOCP can reduce this number for certain types of MRFs).

Quadratic relaxations: L2QP and CQP In [9], the authors approximate the QP with a Convex relaxation (**CQP**) by replacing (W, V) with $(W - diag(D), V + D)$ where $D = W1$ for example makes $W - diag(D) \preceq 0$. The resulting program can be solved in polynomial time. In [10, 11], the constraint $\sum_a x_{ia} = 1$ is relaxed to the L2 constraint $\sum_a x_{ia}^2 = 1$. This L2 relaxation to the QP (**L2QP**) allows for exact optimization of the resulting program when W, V are nonnegative, *even though the problem is non-convex*. The authors map the solution back to the simplex Ω_s before discretizing it.

4 Spectral Relaxation to the Quadratic Program (SQP)

We introduce here our main contribution, which is a Spectral Relaxation to the QP (denoted as **SQP**). One of the fundamental difficulties tackled by all the above methods is the non-convexity of the QP, either in the form of the rank 1 constraint or in the form of the objective. Our work is most closely related to L2QP, in that we still optimize a non-convex cost function, but instead of modifying the *constraint* we modify the *cost function*. The SQP relaxation is defined as follows:

$$\max \epsilon_S(x) = \frac{x^\mathsf{T} W x + V^\mathsf{T} x}{x^\mathsf{T} x + \beta}, \quad \text{s.t.} \quad Cx = 1 \tag{4}$$

where $\beta \geq 0$ is a constant discussed later. Intuitively the normalization $x^\mathsf{T} x$ will encourage "flatter" solutions than without the normalization, thus helping the constraint $0 \leq x \leq 1$ to be satisfied. As we will see, with a good choice of β, one gets a solution that is as "spread out" as possible, while satisfying $0 \leq x \leq 1$.

The advantages of this formulation are three-fold: 1) the optimum of SQP is provably "close" to the optimum of IQP (with exactly the same optimal discrete solutions as we will see). 2) the SQP can be solved very efficiently, inheriting the speed and scalability of spectral methods, and 3) the SQP has a closed form solution in terms of eigenvector of a certain matrix. This property is quite unique, and among other things it would allow one to perform perturbation analysis on the (relaxed) solution. Before we proceed let us give a few more definitions.

Definitions Let $x^* \in \Omega_d$ be an optimal solution of (2) and $\epsilon^* = \epsilon(x^*)$. Let $x_S \in \Omega_a$ be an optimal solution of (4) and $\epsilon_S^* = \epsilon_S(x_S)$. Note, β is implicit in this short-hand notation. Let $E[W]$ denote the average of the elements in W.

4.1 How good is the approximation ?

This question is a central focus of our paper, and we will derive optimality bounds for the relaxed and discretized solutions. Section 5 will improve those bounds by taking into account statistics of the input matrices.

$$\begin{aligned}
&\forall x \in \Omega_d, \quad \tfrac{1}{n+\beta}\epsilon(x) &&= \epsilon_S(x) \\
&\forall x \in \Omega_s, \quad \tfrac{1}{n+\beta}\epsilon(x) &&\leq \epsilon_S(x) \leq \frac{1}{n/k+\beta}\epsilon(x) \\
&\forall x \in \Omega_a, &&\epsilon_S(x) \leq \frac{1}{n/k+\beta}\epsilon(x)
\end{aligned}$$

Proof $\forall x \in \Omega_d, \sum_a x_{ia}^2 = \sum_a x_{ia} = 1$ so $x^\mathsf{T} x = n$. $\forall x \in \Omega_a, 1 = (\sum_a x_{ia})^2 \leq k \sum_a x_{ia}^2$ so $x^\mathsf{T} x \geq n/k$. $\forall x \in \Omega_s, \sum_a x_{ia}^2 \leq \sum_a x_{ia} = 1$ so $x^\mathsf{T} x \leq n \square$

The first equation shows that the IQP and the SQP have the *same optimal discrete solutions*. The second equation shows that on Ω_s, *the SQP approximates the QP by a factor* $\leq \frac{n/k+\beta}{n+\beta}$. The next proposition gives optimality bounds for the MAP problem.

Proposition 4.1 (Data-independant lower bound) *$\epsilon(x_S) \geq \frac{x_S{}^\mathsf{T} x_S + \beta}{n+\beta}\epsilon^* \geq \frac{n/k+\beta}{n+\beta}\epsilon^*$, plotted in figure 1. Corollary: when $x_S \geq 0$, we can efficiently find some $y \in \Omega_d$ with $\epsilon(y) \geq \frac{n/k+\beta}{n+\beta}\epsilon^*$.*

Proof By definition of x_S, $\epsilon_S(x^*) \leq \epsilon_S(x_S)$, leading to the first part with similar arguments as before. The corollary comes from the fact that $x_S \geq 0 \Rightarrow x_S \in \Omega_s$, and we can apply proposition 2.1 $\square$

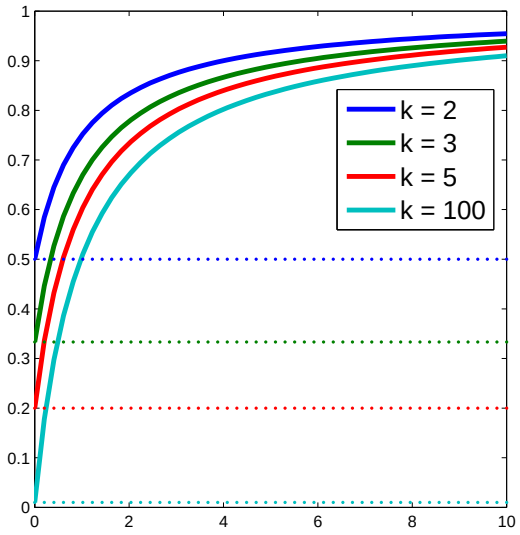

Figure 1: Data independent lower bounds. x-axis: β/n with $n = 100$ (see text). y-axis: lower bound $f(\beta, n, k) = \frac{n/k+\beta}{n+\beta} \geq \frac{\beta}{n+\beta}$ on the ratio $\epsilon(x_S)/\epsilon^*$. The dotted line indicates the corresponding bound $f_{L2QP}(n, k) = \frac{1}{k}$ from [11].

Let β_{max} be the maximal element such that $\beta \leq \beta_{max} \implies x_S \geq 0$ (its existence is discussed later). When β goes from 0 to β_{max}, the lower bound improves because $(z, \beta) \mapsto \frac{z+\beta}{n+\beta}$ increases in both its arguments when $\beta \geq 0, z \in [n/k, n]$, and ${x_S}^\mathsf{T} x_S \in [n/k, n]$ increases with β. Therefore the best bound is obtained for β_{max}. In practice, however, we can tolerate some slack (x_S close to nonnegative), a β slightly superior to β_{max} will result in better discretized solutions.

By the results above, if we could include the constraint $x \geq 0$ to the SQP (4) and increase β, we could get feasible solutions arbitrary close to the optimum, but unfortunately that's NP-hard as we show here:

Theorem 4.2 *(**Solving for eigenvectors under *inequality* constraints is NP-hard**) Let A, B, c be arbitrary matrices of size $nn, mn, m \times 1$. Unless $P = NP$, there is no Polynomial Time Approximation Scheme (PTAS) for the following problem:*

$$\max \frac{x^\mathsf{T} A x}{x^\mathsf{T} x}, \quad s.t. \quad Bx \leq c, \tag{5}$$

Proof see appendix. The proof is constructive and derives a solution to the IQP from the above problem.

Conditions to guarantee $x_S \geq 0$. When W, V are nonnegative and β is small enough w.r.t. $\hat{\beta}$ (defined in section 4.2), for example 0, we observed empirically that $x_S \geq 0$ is almost always satisfied. In fact, $\forall \beta \geq 0$, one can show that $\exists \alpha \in \mathbb{R}$ with that condition being satisfied for $W + \alpha 1 1^\mathsf{T}$. In future work, it would be interesting to find reasonably tight sufficient conditions as well as an estimate of β_{max}.

4.2 Getting the best *upper* bound on ϵ^*

From proposition 4.1, $\epsilon^* \leq \frac{n+\beta}{{x_S}^\mathsf{T} x_S + \beta} \epsilon(x_S)$, giving a family of upper bounds, one for each β. Note, the nonnegativity of x_S is *irrelevant* for getting upper bounds, so we seek the optimal β_{opt} that will *minimize the upper bound*. The following heuristic approximates β_{opt}:

$$\beta_{opt} \approx \hat{\beta} = n^2 E[W] / \epsilon_S^{\bar{W}, 0*} \tag{6}$$

where $\bar{W} = W - E[W] 1 1^\mathsf{T}$ is zero-mean (see Definitions), and $\epsilon_S^{W,\beta}(x) = \frac{x^\mathsf{T} W x + V^\mathsf{T} x}{x^\mathsf{T} x + \beta}$ $\forall W, \beta$. We experimentally justify this expression in the results section. We verified empirically that $\hat{\beta}$ predicts the optimum β_{opt} within a factor 5%.

The fact that we can get both lower bounds *and* upper bounds is a distinguishing feature of our method. We can combine in practice excellent pairs of upper bounds and lower bounds: when W, V are nonnegative, the discrete solution we obtain is typically within a factor > 0.8 of the upper bound we get (with a different β).

5 Data dependent lower bound

We can get improved bounds if we consider the statistics of the input matrices. We follow a similar procedure as in [11]. Suppose, WLOG, that the indexes have been permuted such that the optimal assignment verifies $\forall i, x^*_{i1} = 1$ and 0 otherwise. In this section we assume WLOG that W, V *are nonnegative* (adding a constant to W and V will not change the MAP solution, so we can assume the original MAP problem verified that property). We also introduce matrix $M = W + diag(V)$, which verifies: $\forall x \in \Omega_s, x^\mathsf{T} M x \leq \epsilon(x)$ with *equality* on Ω_d. M has the following block structure:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2:k} \\ {M_{1,2:k}}^\mathsf{T} & M_{2:k,2:k} \end{bmatrix}$$

$M_{1,1}$ corresponds to all the correct assignments, and therefore we notice that $\mathbf{1}^\mathsf{T} M_{1,1} \mathbf{1} = {x^*}^\mathsf{T} M x^* = \epsilon^*$. *Let us introduce* p be the largest element in $[0, 1]$ such that $pE[M_{1,1}] \leq E[M_{2:k,2:k}], pE[M_{1,1}] \leq E[M_{1,2:k}]$, as in [11]. $p \approx 0$ corresponds to a peaked maximum, while $p \approx 1$ corresponds to a more uniform distribution. Such p always exists as we assumed W, V to be nonnegative. We will prove the following property:

Proposition 5.1 (data-dependent lower bound) *$\epsilon(x_s) \geq f(p, k)\epsilon^*$, where $f(p, k) \geq p$ is plotted in figure 2*

We will derive below $f(p,k)$. It's precise expression is a little complicated, but we plot $q \mapsto f(p,k)$ for different values of the number of labels k in figure 2. When k is large, $f(p,k) \sim p$, which implies that $\epsilon(x_s) \geq p\epsilon^*$ regardless of the number of labels.

Comparison with L2QP As figure 2 shows, the bound outperforms the one reported in [11], which was $\epsilon(x_{L2QP}) \geq \frac{1+(k-1)p^2}{k}\epsilon^*$. For k large this only gives $\epsilon(x_{L2QP}) \geq p^2\epsilon^*$. A more careful analysis however shows that L2QP can obtain the same bound as ours (even though we could further increase our bound by taking β into account).

Comparison with CQP It is interesting to compare this bound to the one in [9], which gave an additive bound for their method $\epsilon(x_{CQP})$ The following proposition shows that the bound we obtain is better for most values of p, especially when $k \geq 3$.

Proposition 5.2 (multiplicative bound for CQP) *In the most favorable case for CQP, when p satisfies both $pE[M_{1,1}] = E[M_{2:k,2:k}]$ and $pE[M_{1,1}] = E[M_{1,2:k}]$, the additive bound given in [9] can be transformed into the following multiplicative bound: $\epsilon(x_{CQP}) \geq (\frac{3}{4} - p\frac{k^2-1}{4})\epsilon^*$, also plotted in figure 2.*

Proof of proposition 5.1 By definition, $\forall y \in \Omega_s, \epsilon_S(x_S) \geq \epsilon_S(y)$; we need to find a good $y \in \Omega_s$ which will yield the desired inequality. A natural choice is to consider a y that puts a larger weight to optimal assignments than non-optimal assignments, as uniformly as possible: we can verify that it leads to $y_{i1} = 1/(1+q(k-1))$, and $y_{ia} = q/(1+q(k-1))$ for $a > 1$ (q is a parameter we will adjust). As before we obtain:

$$\epsilon(x_S) \geq \frac{n/k+\beta}{y^\mathsf{T}y+\beta}\epsilon(y) \geq \frac{n/k}{y^\mathsf{T}y}y^\mathsf{T}My$$

because ${x_S}^\mathsf{T}x_S \geq n/k$ and $y^\mathsf{T}y \geq n/k$, and also by definition of M. Using the definition of y, we obtain:

$$\frac{y^\mathsf{T}My}{y^\mathsf{T}y} = \frac{\mathbf{1}^\mathsf{T}M_{1,1}\mathbf{1} + 2q\mathbf{1}^\mathsf{T}M_{1,2:k}\mathbf{1} + q^2\mathbf{1}^\mathsf{T}M_{2:k,2:k}\mathbf{1}}{n(1+(k-1)q^2)}$$

From before, $\mathbf{1}^\mathsf{T}M_{1,1}\mathbf{1} = \epsilon^*$. Let us now use the definition of p and the relative sizes of the blocks in M: the numerator is $\geq \epsilon^* + 2qp(k-1)\epsilon^* + q^2p(k-1)^2\epsilon^*$. Combining everything together, we obtain the bound

$$\frac{\epsilon(x_S)}{\epsilon^*} \geq \frac{q^2p(k-1)^2 + 2qp(k-1)+1}{k(1+(k-1)q^2)}$$

We now find the q^* that maximizes the above expression, and set $f(p,k)$ as the resulting value $q = q^*$. We spare the reader with some tidy calculus and summarize the main result: the above expression has a unique global maximum

$$q^* = \frac{h+\sqrt{4p^2+h^2}}{2(h+1)}, \text{ with } h = p(k-1)-1$$

It satisfies $q^* \in [0,1]$ and $q^* \to 1$ when $k \to \infty$ (corresponding to our intuition), with $f(p,k) \sim p$ at $\infty\square$

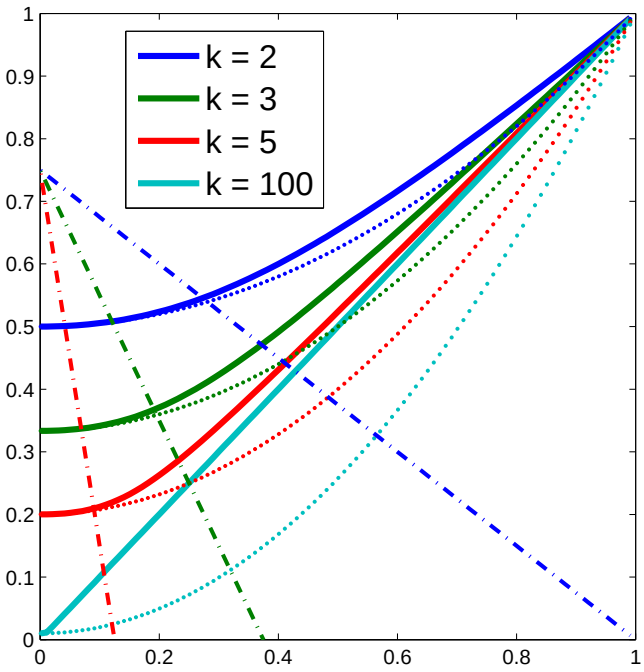

Figure 2: Data-dependent lower bounds. x-axis: $p \in [0,1]$ (see text). y-axis: lower bound $\epsilon(x_S)/\epsilon^* \geq f(p,k)$. Thick plain curves: our algorithm SQP; dotted curves curves: bound published for L2QP[11]; dashed curves: CQP.

Proof of proposition 5.2 see appendix

6 Algorithm and Analysis

6.1 Computational Solution for the SQP

We explain here how to solve (4). For the sake of generality, and since it doesn't change the procedure, let us assume here that W, V, C, b are arbitrary matrices of size resp. $N \times N$, $N \times 1$, $M \times N$ and $M \times 1$ ($M, N \geq 1$). Also, let $\alpha \in \mathbb{R}$ and $\beta > 0$[2]. We solve the following program, *exactly*:

$$\max \quad \frac{x^\mathsf{T}Wx + V^\mathsf{T}x + \alpha}{x^\mathsf{T}x + \beta} \quad \text{s.t.} \quad Cx = b \qquad (7)$$

[2]The case $\beta = 0, V = 0, \alpha = 0, b \neq 0$ is treated with a slightly more complex solution, which we omit for brevity.

Note, the case $\alpha = 0, \beta = 0, V = 0, b = 0$ has been treated in [15, 16]. We give a more general solution here. W.L.O.G, we can assume W symmetric and C full rank. Let us introduce a new variable $t \in \mathbb{R}$, $\bar{x} = [x; t] \in \mathbb{R}^{N+1}$, and the following matrices

$$\bar{W} = \begin{bmatrix} W & \frac{1}{2}V \\ \frac{1}{2}V^\mathsf{T} & \alpha \end{bmatrix}, \bar{D} = \begin{bmatrix} I & 0 \\ 0 & \beta \end{bmatrix}, \bar{C} = \begin{bmatrix} C & -b \end{bmatrix}$$

We verify that (7) is equivalent to:

$$\max \quad \frac{\bar{x}^\mathsf{T}\bar{W}\bar{x}}{\bar{x}^\mathsf{T}\bar{D}\bar{x}} \quad \text{s.t.} \quad \bar{C}\bar{x} = 0 \tag{8}$$

The only non-trivial thing to see is that $\bar{x}^* = [x^*; t^*]$ is an optimum of (8) $\Leftrightarrow$ $[x^*/t^*; 1]$ is an optimum of (8) $\Leftrightarrow$ $x'^* = x^*/t^*$ is an optimum of (7)[3]. Notice the new constraint is *linear* instead of *affine*. Next, we get rid of $\bar{D}$ with a change of variable $x' = \bar{D}^{1/2}\bar{x}$, $W' = \bar{D}^{-1/2}\bar{W}\bar{D}^{-1/2}$, $C' = \bar{C}\bar{D}^{-1/2}$:

$$\max \quad \epsilon_1(x') = \frac{x'^T W' x'}{x'^T x'} \quad \text{s.t.} \quad C'x' = 0, \tag{9}$$

Since C' is full rank as C, we can apply the results of [15, 16], which compute the Lagrangian: the solution to (9) is given by the leading eigenpair of the system

$$P_C W' P_C \quad x' = \lambda x', \tag{10}$$

with $P_C = I - C'^T(C'C'^T)^{-1}C'$.

6.2 Efficient Computation of P_C in the eigensolver

The previous section showed one could reduce (7) to an eigenvector computation. Although the solution described is sufficient for small problem sizes, it is quite inefficient for larger problems, because one needs to invert $C'C'^T$, which is usually a full matrix even if C is sparse. We show here one can do better. We compute $C' = \begin{bmatrix} C & -1/\sqrt{\beta}b \end{bmatrix} = \begin{bmatrix} C & b' \end{bmatrix}$ for some b'. By applying the Sherman-Morrison formula[17], we have:

$$(C'C'^\mathsf{T})^{-1} = (CC^\mathsf{T} + b'b'^\mathsf{T})^{-1} = Z - \frac{Zb'b'^\mathsf{T}Z}{1 + b'^\mathsf{T}Zb'}, \tag{11}$$

where $Z = (CC^\mathsf{T})^{-1}$. In the case of SQP, matters are quite simple, and it is easily shown that $Z = \frac{1}{k}I_n$ if there are k labels per node, and $Z = diag(\frac{1}{k_i})$ if there are k_i labels for node i.

For arbitrary C we can still compute Z efficiently either by computing the QR decomposition of C^T, or by computing the Incomplete Cholesky Descomposition of CC^T. For large problems, we never explicitly form Z, instead we solve two triangular systems at each iteration of an iterative eigensolver.

[3] When $t^* = 0$, (8) has a solution, unlike (7) which only a diverging sequence of points approximates

6.3 Obtaining Discrete Solutions

We need to discretize the continuous solution $x_S \in \Omega_a$ in order to get an approximate solution. If $x_S \not\geq 0$, we map the solution back to the simplex (as one can check) with the following: $x^{(0)} = \frac{1}{k} + \frac{x_S - 1/k}{||x_S - 1/k||_\infty}$. If $x_S \geq 0$ we simply take $x^{(0)} = x_S$. Now we could use the ICM-like construction given in proof of proposition 2.1 to discretize $x^{(0)}$ into some y and still get $\epsilon(y) \geq \epsilon(x^{(0)})$. However, we get better performance with Relaxation Labeling[6] and other related annealing algorithms. For the sake of comparison, we follow (almost exactly) the same discretization procedure as in [11], which gives good results. It is summarized in the next section, along with the rest of our algorithm.

6.4 Summary of the SQP Algorithm

1. Input: clique potentials $W = (W_{iajb}), V = (V_{ia})$, of size $nk \times nk$ and nk.
2. Set $\beta = \hat{\beta}$ using equaton (6)[a]. Compute the first eigenvector x' of $P_C W' P_C$, then $\bar{x}$ and finally x_S, solution of the SQP, as described in sections 6.1 and 6.2.
3. Output upper bound $\epsilon^* \leq \frac{n+\beta}{x_S{}^\mathsf{T} x_S + \beta}\epsilon(x_S)$
4. Initialize $x^{(0)} := x_S$. If $x_S \not\geq 0$, take $x^{(0)} := \frac{1}{k} + \frac{x_S - 1/k}{||x_S - 1/k||_\infty}$
5. Discretization step: set $\theta := \theta_0$ and repeat until convergence
   (a) set $v_{ia} := (Wx^{(t)} + V)_{ia}$
   (b) set $y_{ia} := \exp(\theta v_{ia}) x_{ia}^{(t)}$, and $x_{ia}^{(t+1)} := y_{ia} / \sum_b y_{ib}$
   (c) set $\theta := (1 + \tau)\theta$ after updating $x^{(t+1)}$ for all sites
6. Output $x^{(\text{nbIter})}$, replacing in the last iteration step 5b by a non-maximum suppression.

[a] In practice we use $\beta = \hat{\beta}$ to obtain the upper bound, and a small number (≈ 3) of values $\beta \leq \hat{\beta}$ to obtain discrete solutions

6.5 Computational Cost

The cost of this algorithm is dominated by the computation of the leading eigenvector of (10), which can be computed by the power method or a standard iterative eigensolver such as Lanczos. The cost of each iteration is roughly the cost per matrix-vector operation $x^{t+1} := P_C(W'(P_C x^t))$. From section 6.2 it is easily shown that $y := P_C x^t$ takes $O(N) = O(nk)$ operations, giving a total of $O(nnz(W')) = O(nnz(W)) =$

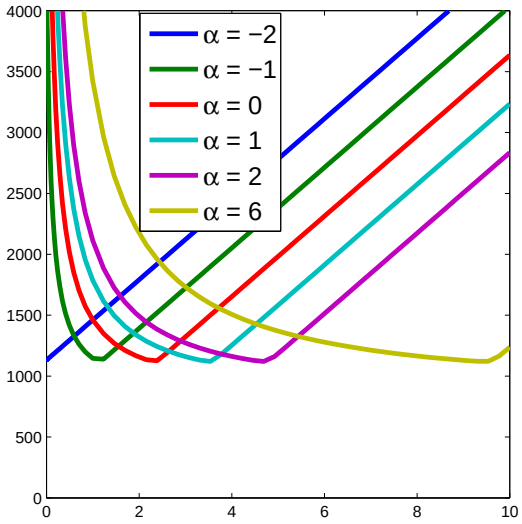

Figure 3: Family of Upper bounds. This plot explains our heuristic for $\hat{\beta}$ in equation (6). W is a random matrix with $E[W] = 2$, $n = 20, k = 10$. x-axis: β/n; y-axis: experimental upper bound. The red curve ($\alpha = 0$) shows the upper bound $\epsilon^* \leq f_{upper}(\beta) = \frac{n+\beta}{x_S{}^\mathsf{T} x_S + \beta} \epsilon_S^{W,\beta *}$ we derived earlier. The optimal upper bound is reached for $\beta = \beta_{opt} \approx 2.2n$. More generally, letting $W_\alpha = W + \alpha 11^\mathsf{T}$, we can also show that $\epsilon^* \leq f'_{upper}(\alpha, \beta) = \frac{n+\beta}{x_S{}^\mathsf{T} x_S + \beta} \epsilon_S^{W_\alpha,\beta *} - \alpha n^2$, and we observe an *affine* relation between α and β_{opt}. From this, we derive our expression for $\hat{\beta}$ by considering the initial conditions $\alpha = -2 = -E[W]$, when $\beta_{opt} \approx 0$.

$O(k^2|E|)$ operations per iteration. In practice, convergence is fast and we can set up a maximum number of iterations, so the total algorithm complexity is *linear* in the problem description length. Note this compares very favorably to other methods discussed so far, and is comparable to the complexity of L2QP.

7 Experiments

7.1 Upper bound computation

We verify experimentally the heuristic expression of $\hat{\beta}$ in equation (6), see figure 3. The plot shows that the upper bound $f_{upper}(\beta)$ is convex in β, and minimized at some $\beta = \beta_{opt}$, which we approximate by noticing the regular spacings of the minima.

7.2 Performance on random MRFs

We compare our SQP algorithm against L2QP, BP, ICM (Iterative Conditional Modes)[12] and Relaxation Labeling. Note, to be fair we use exactly the same discretization procedure for all those methods (it will have no effect on ICM and Relaxation Labeling, since those methods already output a discrete solution). That procedure is described in section 6.4.

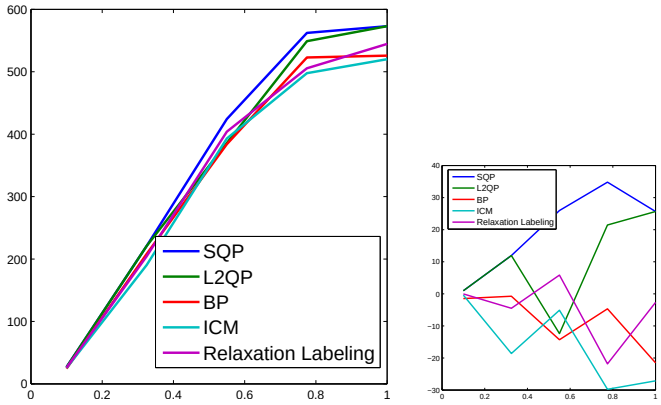

Figure 4: Comparison of different algorithms: SQP (ours) against L2QP, BP, ICM and Relaxation Labeling, see text for details. *Left plot*: x-axis: $p_{edge} \in [0.1..1]$. y-axis: energy output by each algorithm after discretization. Parameters: $n = 50, k = 10$. Results are averaged over 5 iterations per data point. *Right plot*: same as left plot, but for each data point we substract the mean of the energies output by each method, so as to emphasize the differences.

We study the influence of 1) the density of connections in W, and 2) the number of labels k relative to the number of nodes n. For the sake of comparison, we use the same experimental framework as described in [11], which we recall here very briefly. We generate a set of random MRF problems in which we control the density *pedge* of connections in W: $pedge \in [0, 1]$. The potentials (in their exponential form) are drawn uniformly at random with controlled amplitude. We simulate the effect of variable p (section 5) by encouraging connections between pairs of correct labels (set arbitrarily in advance) to be on average larger then other connections. In figure 4, we set $n = 50, k = 10$, and vary p_{edge} between 0.1 and 1 by increments of 0.1. For each value of p_{edge}, we generate 5 random MRFs with the correponding parameters, and compute the energy output by each algorithm after discretization. The results are averaged over each of those 5 trials. In figure 5 we repeat this procedure but with $n = 20, k = 20$. We observe that results are very simiar to those of L2QP, perhaps a little better. ICM performs worse, followed by either BP or Relaxation Labeling.

References

[1] H. Ishikawa. Exact optimization for markov random fields with convex priors, 2003.

[2] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan.

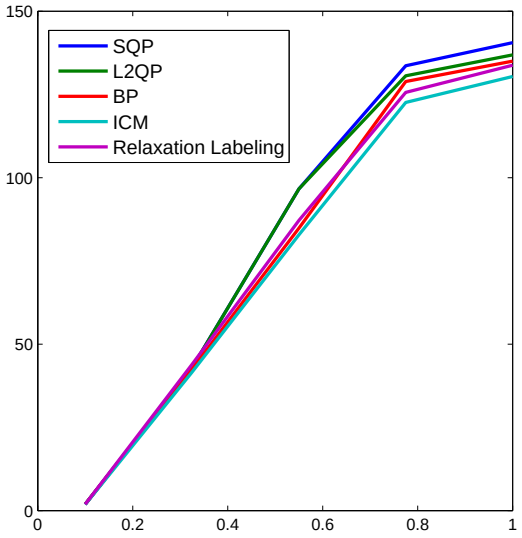

Figure 5: Same caption as figure 4 but with $n = 20, k = 20$.

Loopy belief propagation for approximate inference: An empirical study. pages 467–475.

[3] Weiss and Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEETIT: IEEE Transactions on Information Theory*, 47, 2001.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *ICCV (1)*, pages 377–384, 1999.

[5] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, 2002.

[6] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1983.

[7] Martin J. Wainwright and Michael I. Jordan. Variational inference in graphical models: The view from the marginal polytope.

[8] M. Wainwright, T. Jaakkola, and A. Willsky. Map estimation via agreement on (hyper)trees: messagepassing and linear programming approaches, 2002.

[9] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 737–744, New York, NY, USA, 2006. ACM Press.

[10] Laurent Baratchart, Marc Berthod, and Loïc Pottier. Optimization of positive generalized polynomials under lp constraints. Technical Report RR-2750.

[11] Marius Leordeanu and Martial Hebert. Efficient map approximation for dense energy functions. In *International Conference on Machine Learning 2006*, May 2006.

[12] Julian E. Besag. On the statistical analysis of dirty pictures. In *Journal Of The Royal Statistical Society*, 1986.

[13] P.H.S. Torr. Solving markov random fields using semidefinite programming. *aistats*, 2003.

[14] M.Pawan Kumar, P.H.S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming relaxations. *cvpr*, 1:1045–1052, 2006.

[15] von Matt U. Gander W., Golub G.H. A constrained eigenvalue problem. In *Linear Algebra Appl. 114-115, pp. 815-839*, 1989.

[16] Stella X. Yu and Jianbo Shi. Grouping with bias. In *Advances in Neural Information Processing Systems*, 2001.

[17] Saul A. Teukolsky William T. Vetterling William H. Press, Brian P. Flannery. Numerical recipes in c: The art of scientific computing. In *Cambridge University Press. pp.73+*, 1992.

8 Appendix

Proof of theorem 4.2 Given an arbitrary MAP problem $\epsilon(x) = x^{\mathsf{T}}Wx + V^{\mathsf{T}}x$ (with W, V satisfying the conditions of proposition 2.1) with optimum value $\epsilon^* = \epsilon(x^*)$, there is a finite number of feasible binary assignments, so $\exists \rho > 0 : \forall x, \epsilon(x) \geq \rho\epsilon^* \Rightarrow \epsilon(x) = \epsilon^*$. Let $\rho' < 1, \beta > 0$ be such that $\rho'(k/n + \beta)/(n + \beta) > \rho$. Take z such that $\epsilon_S(z) \geq \rho'\epsilon_S^*$, where $\epsilon_S(x) = \frac{x^{\mathsf{T}}Wx + V^{\mathsf{T}}x}{x^{\mathsf{T}}x + \beta}$ with constraint $Cx = 1, x \geq 0$. Such a z can be found by a related inequality constrained eigenvector problem discussed in the theorem, see section 6.1 for the construction. Since $z \in \Omega_s$ by construction, we can find efficiently a $y \in Omega_d$ such that $\epsilon(y) \geq \epsilon(z) \geq \rho'\epsilon_S^* \geq \rho'(z^{\mathsf{T}}z + \beta)/(n + \beta)\epsilon^* \geq \rho\epsilon^* \Rightarrow \epsilon(y) = \epsilon^*$, giving a polynomial time reduction to solve the MAP from z□

Proof of proposition 5.2 In theorem 3.3 of [9], letting D be the diagonal matrix with elements $d(s; i)$, we can check that $\sum_{s,i} d(s; i) \geq 1^{\mathsf{T}}W1$, because the matrix $W - D$ has to be negative semidefinite. But equality is obtained when taking $D = diag(W1)$ (which is a sufficient choice since we assumed W nonnegative), and so we have $\epsilon(y) \geq \epsilon^* - \frac{1}{4}1^{\mathsf{T}}W1 \geq \epsilon^* - \frac{1}{4}1^{\mathsf{T}}M1$ (since we assumed V nonnegative), so $\epsilon(y) \geq \epsilon^* - \frac{1}{4}(\epsilon^* + 2p(k-1) + p(k-1)^2) = (\frac{3}{4} - p\frac{k^2-1}{4})\epsilon^*$ (similarly to the proof of 5.1)□