2007

Solving Markov Random Fields with Spectral Relaxation

Timothee Cour
University of Pennsylvania

Jianbo Shi
University of Pennsylvania, jshi@cis.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/cis_papers
Part of the Computer Sciences Commons

Recommended Citation

©2007 held by the authors.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_papers/551
For more information, please contact libraryrepository@pobox.upenn.edu.
Solving Markov Random Fields with Spectral Relaxation

Abstract
Markov Random Fields (MRFs) are used in a large array of computer vision and machine learning 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.

Disciplines
Computer Sciences

Comments

©2007 held by the authors.
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 and machine learning 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 interdependent 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[2].

MRFs have been studied extensively since the 1970’s, and a lot of work has been focused on developing approximation algorithms for the MAP problem. Bayesian methods such as Belief Propagation (BP)[3, 4], Generalized BP and Tree Reweighted BP[5] 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[4]. However, there is no general convergence guarantee and BP may fail to converge even in simple graphs. Energy Minimization methods such as Graph Cuts[6, 2] 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, [6] introduces $\alpha - \beta$ swaps and $\alpha$ expansion 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 $O(\text{#edges} \times \text{#labels}^2)$, 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 byproduct, 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[7], Deterministic annealing and LP relaxation[8, 5]. 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 2formulates the MAP-MRF problem and shows how to reduce it to a Quadratic Program (QP). Section 3summarizes the approximation algorithms that try to solve the QP. Section 4 introduces our new Spectral relaxation to the MAP-MRF problem and shows how to reduce it to the MAP-MRF problem and shows how to reduce it 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\(^1\). Let \(G\) be an undirected graph with \(n\) vertices or sites, and edge set \(E\). We attach to each vertex \(i\) a random variable \(X_i \in \{1, ..., k\}\)\(^2\), designing the state of that site. A set of binary and unary potential functions \(\Psi_{ij}\) and \(\Phi_i\) determine compatibility of assignments of neighboring or individual vertices. The joint distribution represented by the MRF is:

\[
P(X) = \frac{1}{Z} \prod_{i,j \in E} \Psi_{ij}(X_i, X_j) \prod_i \Phi_i(X_i),
\]

(1)

where \(Z\) is a constant. The Maximum Apстерiori (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 \(C x = 1\) for a certain matrix \(C\). Next, we introduce the \(nk \times nk\) matrix \(W\) as \(W_{iajb} = \log \Psi_{ij}(a, b)\) if \(ij \in E, W_{iajb} = 0\), and the \(nk \times 1\) vector \(V\) as \(V_{ia} = \log \Phi_i(a)\). WLOG, we can assume \(W\) symmetric. With these notations, log \(P(X) = \sum_{ij \in E} W_{iajb} x_{ia} x_{jb} + \sum_i V_{ia} x_{ia} + \text{constant}\) and the MAP problem becomes:

\[
\max \epsilon(x), \quad \text{s.t.} \; C x = 1, 0 \leq x \leq 1
\]

(2)

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

2.2 QP relaxation

The QP relaxation relaxes the set \(\Omega_d\) to \(\Omega_s\) in (2):

\[
\max \epsilon(x), \quad \text{s.t.} \; C x = 1, 0 \leq x \leq 1
\]

(3)

We extend in the following proposition some recent results from [11, 9], which only considered the case \(W_{iajb} = 0, \forall i, a, b\). Terms of the form \(W_{iajb}, a \neq b\) have no effect on the original IQP, but they affect the solution of the relaxed QP, thus giving us an additional degree of freedom. This proposition will be used later on to prove our optimality bounds.

Proposition 2.1 (QP is equivalent to IQP)

Suppose \(W_{iaia} \geq 2W_{iajb} \forall a \neq b, \forall \text{other entries in } W\) 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, \(\max_{x \in \Omega_d} \epsilon(x) = \max_{x \in \Omega_s} \epsilon(x_d)\) and (3) is equivalent to (2).

In the rest of the paper, we assume WLOG that the condition \(W_{iaia} \geq 2W_{iajb} \forall a \neq b\) is always met, since it is easy to see that terms of the form \(W_{iajb}, a \neq b\) don’t affect the IQP (2).

Proof of proposition 2.1 The proof uses a construction similar to ICM (Iterative Conditional Modes)\(^3\), but requires special treatment for terms of the form \(W_{iajb}\). Let \(y^0 = x\), and, for \(t = 1, n\) let \(y_t = y_{t-1}\) except for \(i = t\): let \(v_d = 2\sum_{(j,b) \neq (t,a)} W_{jaib} y_{t-1}^{ja} + W_{iaa} y_{t-1}^{ia} + V_{ia}\) and \(c = \arg \max_a v_a\). We take \(y_{ta}^t = 1\) and \(y_{ta}^t = 0\) 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 \(2\sum_{b \neq c} W_{tcxb} y_{t-1}^{tc} + W_{tcca} y_{t-1}^{ta} \leq W_{tca}\) because of the hypothesis and the fact that all \(y^t \in \Omega_s\). Finally, we take \(x_d := y^n\), \(x_d \in \Omega_d\). The corollary comes from the fact that \(\max_{x \in \Omega_d} \epsilon \geq \max_{\Omega_s} \epsilon \).

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.

\(^1\)General MRFs can be converted to that form, see [12] \(^2\)It is straightforward to extend our results to the case of site-dependent variable number of labels
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: 
\[ x^T W x + V^T x = \langle X, W_{eq} \rangle \]  
where \( X = [x; 1][x; 1]^T \) is constrained to be rank 1 as well as additional affine constraints. The LP relaxation [8, 5] approximates the non-convex rank 1 constraint or in the form of the fundamental difficulties tackled by all the above methods is the non-convexity of the QP, either in the form of the local consistency constraints. The authors show its relation to tree-reweighted belief propagation, and state conditions for optimality. The SDP relaxation [14] attempts to find a tighter relaxation by approximating \( X = [x; 1][x; 1]^T \) to \( X \succeq [x; 1][x; 1]^T \), but suffers from expensive SDP solvers. The SOCP relaxation [15] proposes a more efficient method than SDP by further relaxing \( X \succeq [x; 1][x; 1]^T \) to \( \langle X, S \rangle \succeq [x; 1]^T S [x; 1] \) for a suitable choice of symmetric matrices \( S \in S \). 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 \( (W - \text{diag}(D), V + D) \) where \( D = W1 \) for example makes \( W - \text{diag}(D) \leq 0 \). 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 \( \Omega_s \) 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:

\[
\max_{\epsilon_S(x)} = \frac{x^T W x + V^T x}{x^T x + \beta}, \quad \text{s.t.} \quad C x = 1
\]

where \( \beta \geq 0 \) is a constant discussed later. Intuitively the normalization \( x^T x \) will encourage “flatter” solutions, helping to enforce the constraint \( x \in [0,1] \). As we will see, a good choice of \( \beta \) will give a solution that is as “spread out” as possible while satisfying \( x \in [0,1] \).

The advantages of this formulation are three-fold: 1) the optimum of SQP is provably close to the optimum of IQP, with 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 would allow one to perform perturbation analysis on the relaxed solution. Before we proceed we need a few more definitions.

Definitions Let \( x^* \in \Omega_s \) be an optimal solution of (2) and \( \epsilon^* = \epsilon(x^*) \). Let \( x_S \in \Omega_s \) be an optimal solution of (4) and \( \epsilon_S^* = \epsilon_S(x_S) \). Note, \( \beta \) 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.

Proposition 4.1 (Initial bounds)

\[
\forall x \in \Omega_d, \quad \frac{1}{n}\epsilon_S(x) = \epsilon_S(x)
\]

\[
\forall x \in \Omega_s, \quad \frac{1}{n + \beta}\epsilon_S(x) \leq \epsilon_S(x) \leq \frac{1}{n/k + \beta}\epsilon(x)
\]

\[
\forall x \in \Omega_s, \quad \epsilon_S(x) \leq \frac{1}{n/k + \beta}\epsilon(x)
\]

Proof \( \forall x \in \Omega_d, \sum a x_{ia}^2 \leq \sum a x_{ia} = 1 \) so \( x^T x = n \). \( \forall x \in \Omega_s, 1 = (\sum a x_{ia})^2 \leq k \sum a x_{ia}^2 \) so \( x^T x \geq n/k \). \( \forall x \in \Omega_s, \sum a x_{ia}^2 \leq \sum a x_{ia} = 1 \) so \( x^T x \leq n \).

The first equation shows that the IQP and the SQP have the same optimal discrete solutions. The second equation shows that on \( \Omega_s \), SQP approximates QP by a factor \( \frac{n/k + \beta}{n + \beta} \). The next proposition states one of our main results, giving bounds for the MAP problem.

Proposition 4.2 (Data-independent lower bound)

\[
\epsilon_S(x_S) \geq \frac{x_S^T x_S + \beta \epsilon^*}{n + \beta} \geq \frac{n/k + \beta}{n + \beta}\epsilon^*, \quad \text{plotted in figure 1.}
\]

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 in proposition 4.1. The corollary comes from \( x_S \geq 0 \Rightarrow x_S \in \Omega_s \), and we can apply proposition 2.1 □
Let $\beta_{\text{max}}$ be the maximal element such that $\beta \leq \beta_{\text{max}} \Rightarrow x_S \geq 0$ (its existence is discussed later). When $\beta$ goes from 0 to $\beta_{\text{max}}$, the lower bound improves because $(z, \beta) \to \frac{z \beta}{n+\beta}$ increases in both its arguments when $\beta \geq 0$, $z \in [n/k, n]$, and $x_S^T x_S \in [n/k, n]$ increases with $\beta$. Therefore the best bound is obtained for $\beta_{\text{max}}$. In practice, however, we can tolerate some slack ($x_S$ close to nonnegative), a $\beta$ slightly superior to $\beta_{\text{max}}$ 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 $\beta$, we could get feasible solutions arbitrary close to the optimum, but unfortunately that’s NP-hard as we show here:

**Theorem 4.3** (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:

$$\max \frac{x^T Ax}{x^T x}, \quad \text{s.t. } Bx \leq c,$$

(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 $\beta$ is small enough w.r.t. $\beta$ (defined in section 4.2), for example 0, we observed empirically that $x_S \geq 0$ is almost always satisfied. In fact, one can show that $\forall \beta \geq 0$, $\exists \alpha \in \mathbb{R}$ s.t. $x_S \geq 0$ when we replace $W$ by $W + \alpha 11^T$. In future work, it would be interesting to find reasonably tight sufficient conditions as well as an estimate of $\beta_{\text{max}}$.

### 4.2 Getting the best upper bound on $\epsilon^*$

From proposition 4.2, $\epsilon^* \leq \frac{n+\beta}{x_S + x_S^2} \epsilon(x_S)$, giving a family of upper bounds, one for each $\beta$. Note, the non-negativity of $x_S$ is irrelevant for getting upper bounds, so we seek the optimal $\beta_{\text{opt}}$ that will minimize the upper bound. The following heuristic approximates $\beta_{\text{opt}}$:

$$\beta_{\text{opt}} \approx \beta = n^2 E[|W|/\epsilon_S]$$

(6)

where $\bar{W} = W - E[|W|]11^T$ is zero-mean (see Definitions), and $\epsilon_S^{W, \beta}(x) = \frac{x^T \bar{W} x + \beta}{x^T x + \beta}$ $\forall W, \beta$. We experimentally justify this expression in the results section. We verified empirically that $\beta$ predicts the optimum $\beta_{\text{opt}}$ 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 $\beta$).

### 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 $\forall i, x'_1 = 1$ 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 $M = W + \text{diag}(V)$, which verifies: $\forall x \in \Omega, x^T M x \leq \epsilon(x)$ with equality on $\Omega_d$. $M$ has the following block structure:

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

$M_{1,1}$ corresponds to all the correct assignments, and therefore we notice that $1^T M_{1,1}1 = x^* M x^* = \epsilon^*$. Let us introduce $p$ be the largest element in $[0, 1]$ such that $p E[M_{1,1}] \leq E[M_{2:k,2:k}], p E[M_{1,1}] \leq E[M_{1,2:k}]$, 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 \( f(p, k) \). Its precise expression is a little complicated, but we plot \( q \rightarrow f(p, k) \) for different values of the number of labels \( k \) in figure 2. When \( k \) is large, \( f(p, k) \sim p \), which implies that \( \epsilon(x_s) \geq pe^* \) regardless of the number of labels.

**Comparison with L2QP** As figure 2 shows, the bound outperforms the one reported in [11], which was \( \epsilon(x_L2QP) \geq \frac{1 + k - 1}{k}p^2e^* \). For \( k \) large this only gives \( \epsilon(x_L2QP) \geq p^2e^* \). A more careful analysis would show that L2QP can obtain the same bound as ours, but we could further increase our bound by taking \( \beta \) into account as we did in the previous section.

**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 multiplicative bound given in [9] can be transformed into the following multiplicative bound:

\[
\epsilon(x_{CQP}) \geq \left( \frac{1}{2} - p\frac{k^2 - 1}{1 + (k - 1)q^2} \right)e^*, \text{ also plotted in figure 2.}
\]

**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 \( y_{i_1} = 1/(1 + q(k - 1)) \), and \( y_{a} = q/(1 + q(k - 1)) \) for \( a > 1 \) (\( q \) is a parameter we will adjust). As before we obtain:

\[
\epsilon(x_S) \geq \frac{n/k + \beta}{y^Ty + \beta} \geq \frac{n/k}{y^TM_y}
\]

because \( x_S^Tx_S \geq n/k \) and \( y^Ty \geq n/k \), and also by definition of \( M \). Using the definition of \( y \), we obtain:

\[
y^TM_y \frac{y^Ty}{y^Ty} = \frac{1^TM_{1,1}1 + 2q1^TM_{1,2,k}1 + q^21^TM_{2,k,2,k}1}{n(1 + (k - 1)q^2)}
\]

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

\[
\frac{\epsilon(x_S)}{\epsilon^*} \geq q^2p(k - 1)^2 + 2pq(k - 1) + 1
\]

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

\[
q^* = h + \sqrt{4p^2 + h^2} \quad \text{with} \quad h = p(k - 1) - 1
\]

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

**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 \( N \times N, N \times 1, M \times N \) and \( M \times 1 \) \((M, N \geq 1)\). Also, let \( \alpha \in \mathbb{R} \) and \( \beta > 0^3 \). We solve the following program, exactly:

\[
\max \quad x^TWx + V^Tx + \alpha \quad \text{s.t.} \quad Cx = b
\]

\( ^3 \)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 [16, 17]. 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

$$W = \left[ \begin{array}{c} W \frac{1}{2} V^T \\ \frac{1}{2} V \end{array} \right], \bar{D} = \left[ \begin{array}{c} I \ 0 \\ 0 \ \beta \end{array} \right], \bar{C} = [ \ C \ -b ]$$

We verify that (7) is equivalent to:

$$\max \ \bar{x}^T W \bar{x} \ \text{s.t.} \ \bar{C} \bar{x} = 0$$

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). Notice the new constraint is linear instead of affine. Next, we get rid of $\bar{D}$ with a change of variable $x' = \bar{D}^{1/2} \bar{x}$, $W' = \bar{D}^{-1/2} W \bar{D}^{-1/2}$, $C' = \bar{C} \bar{D}^{-1/2}$:

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

Since $C'$ is full rank as $C$, we can apply the results of [16, 17], which compute the Lagrangian: the solution to (9) is given by the leading eigenpair of the system

$$P_C W' P_C \ x' = \lambda x'$$

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

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 $C' = \begin{bmatrix} C & -1/\sqrt{3} b' \end{bmatrix} = \begin{bmatrix} C & b' \end{bmatrix}$ for some $b'$. By applying the Sherman-Morrison formula[18], we have:

$$(C'C'T)^{-1} = (CCT + b'b'T)^{-1} = Z - \frac{Zb'b'^T Z}{1 + b'^T Zb'},$$

where $Z = (CCT)^{-1}$. 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 $Z = \text{diag}(\frac{1}{k})$ 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 Decomposition of $CCT$. For large problems, we never explicitly form $Z$, instead we solve two triangular systems at each iteration of an iterative eigensolver.

6.3 Obtaining Discrete Solutions

We need to discretize the continuous solution $x_S \in \Omega_\alpha$ in order to get an approximate solution. If $x_S \not\in \Omega_\alpha$, we map the solution back to the simplex $\Omega_\alpha$ with the following: $x_S^{(0)} = \frac{1}{k} + (x_S - 1/k)/\delta, \delta = \max \left\{ (kx_S^{\text{max}} - 1)/(k - 1), (1 - kx_S^{\text{min}}) \right\}$, as one can check. If $x_S \geq 0$ we simply take $x_S^{(0)} = x_S$. Now we could use the ICM-like construction given in proof of proposition 2.1 to discretize $x_S^{(0)}$ into some $y$ and still get $\epsilon(y) \geq \epsilon(x_S^{(0)})$. However, we get better performance with Relaxation Labeling[7] 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 equation (6). 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 $\epsilon^* \leq \frac{n + \beta}{x_S^{(0)} x_S^{(0)} + \beta} \epsilon(x_S)$
4. Initialize $x^{(0)} := x_S$. If $x_S \not\in \Omega_\alpha$, take $x^{(0)} := 1/k + (x_S - 1/k)/\delta$
5. Discretization step: set $\theta := \theta_0 (\theta_0 = 0.01)$ and repeat until convergence
   (a) set $v_{ia} := (W x^{(t)} + V)_{ia}$
   (b) set $y_{ia} := \exp(\theta v_{ia}) x^{(t)}_{ia}$, and $x^{(t+1)}_{ia} := y_{ia}/\sum_b y_{ib}$
   (c) set $\theta := (1 + \tau) \theta$ after updating $x^{(t+1)}$ for all sites ($\tau = 0.05$)
6. Output $x^{(\text{final})}$, replacing in the last iteration step 5b by a non-maximum suppression.

*In practice we use $\beta = \hat{\beta}$ to obtain the upper bound, and a small number ($\approx 3$) 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
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: $\beta/n$; y-axis: experimental upper bound. The red curve ($\alpha = 0$) shows the upper bound $\epsilon^* \leq f_{\text{upper}}(\beta) = \frac{n+\beta}{n+2+\beta} W_{\alpha,\beta}^*$ we derived earlier. The optimal upper bound is reached for $\beta = \beta_{\text{opt}} \approx 2.2n$. More generally, letting $W_{\alpha} = W + \alpha 11^T$, we can also show that $\epsilon^* \leq f_{\text{upper}}(\alpha, \beta) = \frac{n+\beta}{n+2+\beta} W_{\alpha,\beta} - \alpha n^2$, and we observe an affine relation between $\alpha$ and $\beta_{\text{opt}}$. From this, we derive our expression for $\beta$ by considering the initial conditions $\alpha = -2 = -E[W]$, when $\beta_{\text{opt}} \approx 0$.

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

7 Experiments

7.1 Upper bound computation

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 $p_{\text{edge}}$ of connections in $W$: $p_{\text{edge}} \in [0,1]$. 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 nodes with the 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 $p_{\text{edge}}$ between 0.1 and 1 by increments of 0.1. For each value of $p_{\text{edge}}$, we generate 5 random MRFs with the corresponding 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 similar to those of L2QP, perhaps a little better. ICM performs worse, followed by either BP or Relaxation Labeling.

7.2 Performance on random MRFs

We compare our SQP algorithm against L2QP, BP, ICM (Iterative Conditional Modes)[13] 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.
References


8 Appendix

Proof of theorem 4.3 Given an arbitrary MAP problem $\epsilon(z) = x^T W x + V^T x$ (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 $\exists p > 0 : \forall x \epsilon(x) \geq \epsilon^* \Rightarrow \epsilon(x) = \epsilon^*$. Let $\rho' \in (1, \beta)$ be such that $\rho'(k/n + \beta) \geq \beta > \rho$. Take $z$ such that $\epsilon_S(z) \geq \rho' \epsilon_S^*$, where $\epsilon_S(z) = \frac{z^T W x + z^T V x}{z^T W z + z^T V z}$ with constraint $C x = 1, x \geq 0$. 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 $z \in \Omega$, by construction, we can find efficiently a $y \in \Omega_d$ such that $\epsilon(y) \geq \epsilon(z) \geq \rho' \epsilon_S^* \geq \rho'(z^T z + \beta)/(n + \beta) \epsilon^* \geq \rho' \epsilon^* \Rightarrow \epsilon(y) = \epsilon^*$, 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 $\sum_{s,i} d(s; i) \geq 1^T W 1$, because the matrix $W - D$ has to be negative semidefinite. But equality is obtained when taking $D = diag(W 1)$ (which is a sufficient choice since we assumed $W$ nonnegative), and so we have $\epsilon(y) \geq \epsilon^* - 1^T W 1 \geq \epsilon^* - \frac{1}{4} 1^T W 1$ (since we assumed $W$ nonnegative), so $\epsilon(y) \geq \epsilon^* - \frac{1}{4} (\epsilon^* + 2p(k - 1) + p(k - 1)^2) = \left(\frac{1}{4} - p^{k-1} \frac{k-1}{k}\right) \epsilon^*$ (similarly to the proof of 5.1).