Approximate Inference by Intersecting Semidefinite Bound and Local Polytope

Jian Peng, Tamir Hazan, Nathan Srebro, Jinbo Xu
Toyota Technological Institute at Chicago, {pengjian,tamir,nati,j3xu}@ttic.edu

Abstract

Inference in probabilistic graphical models can be represented as a constrained optimization problem of a free-energy functional. Substantial research has been focused on the approximation of the constraint set, also known as the marginal polytope. This paper presents a novel inference algorithm that tightens and solves the optimization problem by intersecting the popular local polytope and the semidefinite outer bound of the marginal polytope. Using dual decomposition, our method separates the optimization problem into two subproblems: a semidefinite program (SDP), which is solved by a low-rank SDP algorithm, and a free-energy based optimization problem, which is solved by convex belief propagation. Our method has a very reasonable computational complexity and its actual running time is typically within a small factor (≤10) of convex belief propagation. Tested on both synthetic data and a real-world protein side-chain packing benchmark, our method significantly outperforms tree-reweighted belief propagation in both marginal probability inference and MAP inference. Our method is competitive with the state-of-the-art in MRF inference, solving all protein tasks solved by the recently presented MPLP method [19], and beating MPLP on lattices with strong edge potentials.

1 Introduction

Probabilistic graphical models are widely used for reasoning about complex distributions and for modeling problems in computer vision, computational biology and many other real-world applications. Graphical models provide a factorized representation of the complex distribution into local potential functions according to graph structures. Two different inference tasks for graphical models have been widely used and extensively studied. One is to find the assignment of all variables that jointly maximize the probability defined by the model. This is often referred to as the maximum a-posteriori (MAP) assignment. The other is to obtain marginal probabilities of a given set of variables. For arbitrary graphs, both tasks are computationally challenging as they may require summation of, or enumeration over exponential number of assignments.

Although exact inference problems are known to be NP-hard, various empirically successful approximate algorithms have been suggested. Many algorithms for both inference tasks can be unified in the framework of convex optimization with the variational principles. There are two key ingredients in these algorithms, a convex surrogate of the free energy and a tractable convex outer bound of marginal polytope. Several convex surrogates such as tree-weighted free energy and convex free energy have been studied [24, 8]. Various tractable convex outer bounds, such as local polytope and semidefinite outer bound, have been proposed. Based on the local polytope, several efficient message passing algorithms have been studied and achieved great success in practice, such as convex belief propagation, MPLP and tree-weighted belief propagation [24, 8, 6]. However the local polytopes are not tight for hard MAP inference tasks and many other cases of interest. Semidefinite outer bounds have also been proposed for MAP inference [21] and for log-determinantal entropy approximation with binary Markov random fields [22]. Semidefinite programming (SDP) algorithms have great potential in solving hard graphical models, especially for MAP inference where no integral solution of linear programming re-

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

laxation could be found. A large number of SDP-based rounding algorithms have also been studied in theoretical computer science. They are proved to be tighter than linear programming relaxation and effective for many hard combinatorial optimization problems, such as MAXCUT, unique games and binary quadratic programming. Interestingly, these three problems can all be represented as MAP inference problems in graphical models. Empirically, a major issue for SDP approaches is the expensive computational cost of solving these convex optimization problems. Most interior-point-based algorithms for SDPs cannot scale well to graphs with hundreds of nodes, which makes this approach impractical for many real-world applications.

The main contribution of this paper is to propose an approximate inference algorithm for probabilistic graphical models that combines convex belief propagation and semidefinite programming in a scalable way. Our algorithm uses the intersection between the local polytope and the semidefinite constraint as a strong outer bound of the marginal polytope. In addition, our algorithm employs a dual decomposition technique to separate the hard optimization problem into two slave problems: a convex free-energy functional optimization and a semidefinite programming (SDP) optimization with a small number of linear constraints. The convex free-energy functional optimization is solved efficiently by convex belief propagation and the SDP problem is solved using a low-rank SDP algorithm with low computational costs. Like convex belief propagation or message passing algorithms for linear programming relaxations, this low-rank SDP algorithm is able to take advantage of the sparsity of underlying graph structures, thus making the optimization substantially more efficient. We also study several rounding schemes for MAP inference based on the low-rank SDP solution. Tested on both synthetic data and a real-world protein side-chain packing benchmark, our algorithm significantly outperforms tree-reweighted belief propagation for both marginal probability inference and MAP inference. Our algorithm also outperforms the pure SDP method for MAP inference, i.e. our synergised algorithm is indeed better then each of its parts. With SDP rounding schemes, our algorithm can solve all the instances in the protein side-chain packing benchmark to optimal solutions. Our algorithm is comparable to the state-of-the-art method on the protein side-chain packing problem and better on a difficult synthetic data set. Our algorithm has a very reasonable running time, within a small factor (usually ≤ 10) of convex belief propagation.

2 Related Work

Current popular inference algorithms including tree-weighted belief propagation [25, 24] and convex belief propagation [8] are based on the local polytope. With a convex surrogate of the free energy, the algorithms have nice convergence guarantees and are often computational inexpensive since they exploit the structure of the underlying graphs. However, message passing algorithms still suffer from the looseness of local polytope, especially for challenging MAP inference tasks. To tighten the local polytope, high-order polytope has been proposed. In particular, the adaptive-augmented Kikuchi proposed in [19] for MAP inference has been proved to be effective empirically. Coupling with a message passing algorithm from linear programming relaxation, this method gradually tightens the local polytope to a partial Kikuchi polytope with triplets. This method is practically efficient, since LP-based message passing scheme can take advantage of the sparsity of the underlying graphs.

Another direction for MAP inference is to use different outer bounds (i.e., relaxations) of the marginal polytope, e.g., semidefinite outer bound. Semidefinite programming (SDP) has been used for both marginal and MAP inference tasks [22, 21]. SDP-based rounding algorithms have also been extensively studied for hard combinatorial optimization problems, such as MAXCUT, unique games, constraint satisfaction programs [7, 15, 16], and demonstrated stronger theoretical performance than LP-based rounding. Although SDP has been shown to be superior over message passing algorithm with local polytope, SDP usually has an expensive computational cost, which prevents it from being widely used in practice. Other relaxations have also been proposed, such as quadratic programming and second order cone programming [18, 13] for MAP inference. Although these relaxations can be solved much more efficiently than SDP, unfortunately [11] they are no better than LP relaxation for MAP inference.

Very recently, low-rank techniques have been proposed to efficiently solve SDP [4], by substituting the positive semidefinite matrix with a low-rank factorization. This technique has already been applied to many machine learning problems including clustering, embedding and collaborative filtering[3, 12, 10]. We combine this low-rank technique with convex belief propagation so that our inference algorithm can solve the optimization problem very efficiently. In particular, with the solution of the low-rank technique, SDP rounding schemes can be directly applied to MAP inference.

3 Problem Setting

Consider a joint distribution over a pairwise discrete undirected graphical model, or Markov random fields,

$$q(x_1, x_2, ..., x_n) = \frac{1}{Z} \prod_i \phi_i(x_i) \prod_{(i,j)\in E} \psi_{i,j}(x_i, x_j) \quad (1)$$

where ϕ_i and $\psi_{i,j}$ are the potential functions over node i and edge $(i, j) \in E$; Z is the partition function. With the variational principle, we seek to find a distribution p to minimize the KL-divergence between p and q. By expanding $KL(p \parallel \prod_i \phi_i \prod_{i,j\in E} \psi_{i,j})$, our objective is to find the minimizer of an energy functional or free energy,

$$\begin{aligned} F(p) &= -H(p) + \sum_{i,x_i} \theta_i(x_i)p(x_i) \\ &\quad + \sum_{(i,j)\in E,x_i,x_j} \theta_{i,j}(x_i, x_j)p(x_i, x_j) \end{aligned}$$

The term $H(p) = -\sum_x p(x) \ln p(x)$ is the entropy and $\theta_i = -\ln \phi_i$ and $\theta_{i,j} = -\ln \psi_{i,j}$. By minimizing $F(p)$ over the probability simplex $\mathcal{P} = \{\mathbf{p} : \mathbf{p} \geq 0, \sum_x \mathbf{p}(x) = 1\}$, we obtain the actual distribution $p^* = q$ and $-\ln Z = F(p^*)$. The probability simplex $\mathcal{P}$ is also known as the marginal polytope. The optimization problem has a unique optimal solution, since both $F(p)$ and $\mathcal{P}$ are convex.

The Fractional Free Energy and The Local Polytope

When the graph has cycles, the entropy term $H(p)$ is computationally intractable. The satisfaction of the marginal polytope $\mathcal{P}$ is also intractable. The widely-used approximation methods for this optimization problem are based on (1) decomposition of $H(p)$ into local entropy terms; and (2) approximation of $\mathcal{P}$ by simpler convex outer bounds, such as local marginal consistency constraints. The true marginal distributions $p(x_i)$ and $p(x_i, x_j)$ are replaced by "belief" $b_i(x_i)$ and $b_{i,j}(x_i, x_j)$. The global entropy is decomposed into local terms involved with nodes and edges. Fractional free energy with entropy approximation has the form:

$$\begin{aligned} &\sum_{i,x_i} \theta_i(x_i)p(x_i) + \sum_{(i,j)\in E} \sum_{x_i,x_j} \theta_{i,j}(x_i, x_j)p(x_i, x_j) \\ &- \sum_{(i,j)\in E} c_{i,j}H(b_{i,j}) - \sum_i c_i H(b_i) \end{aligned}$$

c_i and $c_{i,j}$ are counting numbers of local entropy terms. For trees, the setting of $c_{i,j} = 1$ and $c_i = 1 - d_i$ where d_i is the degree of node i, is exact and known as the Bethe free energy.

The probability simplex is replaced by a local polytope $\mathcal{L}(b)$ defined below:

$$\mathcal{L}(b) = \begin{cases} \sum_{x_j} b_{i,j}(x_i, x_j) = b_i(x_i), \forall (i, j) \in E \\ b_{i,j}(x_i, x_j) \geq 0, \sum_{x_i,x_j} b_{i,j}(x_i, x_j) = 1 \end{cases}$$

Moreover, when the underlying graph is a tree, the local polytope is equal to the probability simplex or marginal polytope. As a result, the Bethe free energy problem is both exact and convex for a tree-structured graph.

For general graphs with cycles, the Bethe entropy is an approximation of the true entropy. Also, the local polytope is an outer bounds of the marginal polytope. So there is not guarantee of the minimizer of the Bethe free energy problem. From the optimization point of view, the Bethe free energy is no longer convex for a graph with cycles. As a result, the fixed point of the sum-product algorithm is only a local minima of the optimization problem.

By some clever ways of setting counting numbers c, the fractional free energy can be convex [9, 14]. In this work, we assume that we have a set of $c_i, c_{i,j}$ and a nonnegative constant ϵ such that the optimization problem is convex.

$$\begin{aligned} \min_{b\in\mathcal{L}(b)} \quad & \sum_{i,x_i} \theta_i(x_i)b(x_i) + \sum_{(i,j)\in E,x_i,x_j} \theta_{i,j}(x_i, x_j)b(x_i, x_j) \\ & -\epsilon \sum_{(i,j)\in E} c_{i,j}H(b_{i,j}) - \epsilon \sum_i c_i H(b_i) \end{aligned}$$

If $\epsilon = 1$, we obtain the inference problem for marginal probability estimation. By taking $\epsilon \to 0$, the problem becomes MAP inference, whose solution is the joint assignment of all variables x_i such that the probability defined by the model is maximized.

Semidefinite Outer Bound

Assume for each node i, $x_i \in 1, 2, ..., m$. Then MRF can be seen as a distribution over a $(n \times m)$-dimensional binary vector. Because the covariance matrix of this binary vector is positive semidefinite, it is not hard to show that the marginal polytope $\mathcal{M}(b)$ is contained within a semidefinite constraint set $\{b \in R^{nm+nm(nm-1)/2} : M(b) \succeq 0\}$ [22]. See the detailed definition in 2. $M(b) \succeq 0$ is equivalent to the condition of semidefinite positiveness of a covariance matrix by Schur's complement theorem. For convenience, the row and columns of $M(b)$ are indexed by $\{0\} \cup \{(i, x_i)\}$. It is worth noting that $b_{i,j}(s, u) = b_{j,i}(u, s)$ and for each node i and assignment $s \neq t$, the cross term $b_{i,i}(s, t)$ is always zero and $b_{i,i}(s, s) = b_i(s)$, since there can be only a unique assignment to the variable associated with this node.

4 Intersecting local polytope and semidefinite outer bound

In [23], it has been shown that the semidefinite outer bound and local marginal consistency polytope are incomparable. That is, neither semidefinite outer bound nor local polytope is tighter than the other one. There-

$$M(b) = \begin{pmatrix} 1 & b_1(1) & \cdots & b_i(s) & \cdots & b_j(t) & \cdots & b_n(m) \\ b_1(1) & b_1(1) & \cdots & b_{1,i}(1,s) & \cdots & b_{1,j}(1,t) & \cdots & b_{1,n}(1,m) \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ b_i(s) & b_{i,1}(s,1) & \cdots & b_i(s) & \cdots & b_{i,j}(s,t) & \cdots & b_{i,n}(s,m) \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ b_j(t) & b_{j,1}(t,1) & \cdots & b_{j,i}(t,s) & \cdots & b_j(t) & \cdots & b_{j,n}(t,m) \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ b_n(m) & b_{n,1}(m,1) & \cdots & b_{n,i}(m,s) & \cdots & b_{n,j}(m,t) & \cdots & b_n(m) \end{pmatrix} \tag{2}$$

fore, we expect to have a tighter outer bound by intersecting them. We add the semidefinite constraint into the standard convex free energy functional optimization problem as follows.

$$\begin{aligned} \min_b \quad & \sum_{i,x_i} \theta_i(x_i) b(x_i) + \sum_{(i,j)\in E} \sum_{x_i,x_j} \theta_{i,j}(x_i,x_j) b(x_i,x_j) \\ & -\epsilon \sum_{(i,j)\in E} c_{i,j} H(b_{i,j}) - \epsilon \sum_i c_i H(b_i) \\ s.t. \quad & b \in \mathcal{L}(b), M(b) \succeq 0 \end{aligned}$$

For notational simplicity, we use $F(b)$ denote the objective in the above optimization problem.

Although the optimization problem is convex, complicated constraints and nonlinear objective function make it very challenging to solve. Interior-point methods or conditional gradient methods fail to solve even moderate sized instances of this problem with reasonable runtime.

Dual Decomposition

Here we present a dual decomposition approach to separate the semidefinite constraint from the local marginal polytope. The dual decomposition methods for linear programming relaxations can be found in [20]. Typically, in dual decomposition, the slave problems and the master problem have to be defined according to the Lagrangian dual of the original optimization problem. Separate optimizations of the slave problems are assumed to be easy. The master problem will act as the controller or coordinator to pass information among slave problems.

Introducing a set of auxiliary variables, we transform the above optimization problem into:

$$\begin{aligned} \min\nolimits_{b^{(1)}\in\mathcal{L}(b),b^{(2)}} \quad & F(b^{(1)}) + G(b^{(2)}) \\ s.t. \quad & b^{(1)} = b^{(2)} \end{aligned}$$

Here G is an indicator function of the semidefinite constraint, which is also convex.

$$G(b) = \begin{cases} 0, & \exists B \succeq 0, B_{0,0} = 1, \\ & \forall i, x_i \ b_i(x_i) = B_{i,i}(x_i) = B_{0,i}(x_i), \\ & \forall i,j,x_i,x_j \ b_{i,j}(x_i,x_j) = B_{i,j}(x_i,x_j) \\ \infty, & \text{otherwise} \end{cases}$$

where each row and column of matrix B is indexed by $\{0\} \cup \{(i,x_i)\}$. It can be shown that this reformulated problem is equivalent to the original optimization problem.

The Lagrangian dual $L(v)$ of the above problem is as follows.

$$\min_{b^{(1)}\in\mathcal{L}(b),b^{(2)}} \{F(b^{(1)}) + G(b^{(2)}) + v^T(b^{(1)} - b^{(2)})\} =$$

$$\min_{b^{(1)}\in\mathcal{L}(b)} \{F(b^{(1)}) + v^T b^{(1)}\} + \min_{b^{(2)}} \{G(b^{(2)}) - v^T b^{(2)}\}$$

Therefore, two slave problems are defined as $f_1(v) = \min_{b^{(1)}\in\mathcal{L}(b)}\{F(b^{(1)}) + v^T b^{(1)}\}$, and $f_2(v) = \min_{b^{(2)}}\{G(b^{(2)}) - v^T b^{(2)}\}$. The master problem over the dual variable v is $L(v) = \max_v\{f_1(v) + f_2(v)\}$, which is convex and can be solved by projected subgradient method. Assume that $b^{(1)*}$ and $b^{(2)*}$ are the solutions of $f_1(v)$ and $f_2(v)$ respectively. It is easy to see that the subgradient of the master problem is $\nabla L_v = b^{(1)*} - b^{(2)*}$. In this way, we decompose the difficult optimization problem into two easier slave subproblems.

The overall projected subgradient algorithm is shown in Algorithm 1. α_t is the step size of subgradient, which could be set in a number of ways. We use the same scheme proposed in [17], which adjusts the step size according to the primal-dual gap and works well in practice. The algorithms for the two slave problems are described in the next subsections.

Convex Belief Propagation

The first slave problem $f_1(v) = \min_{b^{(1)}\in\mathcal{L}(b)}\{F(b^{(1)}) + v^T b^{(1)}\}$ is the standard convex free-energy functional optimization problem in variational inference, which can be solved by many algorithms, e.g. convex belief propagation, MPLP and tree-reweighted belief propagation. In this work we use the convex belief propa-

```
Initialization: $v = 0$
while not converged do
    1. Obtain $b_t^{(1)*} = \arg\min_b F(b) + v^T b$ by convex belief propagation algorithm;
    2. Obtain $b_t^{(2)*} = \arg\min_b G(b) - v^T b$ by low-rank SDP algorithm;
    3. Update $\alpha_t$;
    4. Update $v = v + \alpha_t(b_t^{(1)*} - b_t^{(2)*})$;
end
Output: $b^* = 0.5(b^{(1)*} + b^{(2)*})$
```

Algorithm 1: Dual Decomposition

gation algorithm to optimize with local beliefs b_i and pairwise beliefs $b_{i,j}$, as shown in Algorithm 2.

```
1) Set $\hat{c}_i = c_i + \sum_{j \in N(i)} c_{i,j}$ , $\hat{\theta} = \theta + v$
2) For every $i = 1, ..., n$ repeat until convergence:
$\forall j \in N(i), x_i$:
    1. $\mu_{j \to i}(x_i) = \ln \sum_{x_j} \exp\left\{\frac{\hat{\theta}_{i,j}(x_i,x_j)+\lambda_{j \to i}(x_j)}{\epsilon c_{i,j}}\right\}^{\epsilon c_{i,j}}$
    2. $\lambda_{i \to j}(x_i) = \frac{c_{i,j}}{\hat{c}_i}\left(\hat{\theta}_i(x_i) + \sum_{k \in N(i)} \mu_{k \to i}(x_i)\right)$
       $-\mu_{j \to i}(x_i)$
3) Obtain $b_i$ and $b_{i,j}$:
    $b_i(x_i) \propto \exp\left\{\frac{\hat{\theta}_i(x_i)+\sum_{j \in N(i)} \mu_{j \to i}(x_i)}{\epsilon \hat{c}_i}\right\}$
    $b_{i,j}(x_i, x_j) \propto \exp\left\{\frac{\hat{\theta}_{i,j}(x_i,x_j)+\lambda_{i \to j}(x_j)+\lambda_{j \to i}(x_i)}{\epsilon c_{i,j}}\right\}$
```

Algorithm 2: Convex Belief Propagation

The Low-rank Algorithm for SDP

The second slave problem $f_2(v) = \min_{b^{(2)}}\{G(b^{(2)}) - v^T b^{(2)}\}$ is equivalent to the following standard SDP:

$$\min_B \sum_{i,x_i} v_i(x_i) B_{i,i}(x_i, x_i) + \sum_{i,j,x_i,x_j} v_{i,j}(x_i, x_j) B_{i,j}(x_i, x_j)$$

$$s.t. \quad B \succeq 0, B_{0,0} = 1, \forall i, x_i \; B_{i,i}(x_i, x_i) = B_{0,i}(x_i)$$

This SDP has a linear objective and a fairly small set of $nm + 1$ linear constraints. However, standard SDP solvers cannot scale well to large nm (i.e., several hundreds). The major reason is that they require representing the complete positive semidefinite matrix and thus, cannot exploit the sparsity of the underlying graph structure. This is similar to the type of structure found in the LP resulting from the local marginal relaxation, which is not exploited by generic methods, but can be exploited by some specially-designed message passing or belief propagation algorithms [26]. In this work, we use a low-rank approach to solve the second slave problem. Specifically, we use gradient updates on a low rank factorization, which takes advantage of the graph sparsity, much in the same way as belief propagation does.

The key idea is to substitute $B = yy^T$ into the above optimization problem, where y is a $(nm + 1) \times r$ matrix and r is the approximation rank. r can be much smaller than $nm + 1$. This approach transforms the original SDP into a quadratic programming optimization problem. This transformation avoids the positive semidefinite constraint on B, at the cost of a resulting problem which is not convex. Nevertheless, at least if r is sufficiently large, there are no local minima [5], and even for moderate r it seems that the higher dimensional relaxation helps avoid local minima (see, e.g. [12]). We then use the augmented Lagrangian technique to solve this quadratic optimization problem.

$$\begin{aligned} L(y, \lambda, \gamma) &= \textstyle\sum_{i,x_i} v_i(x_i) y_i(x_i)^T y_i(x_i) \\ &+ \textstyle\sum_{i,j,x_i,x_j} v_{i,j}(x_i, x_j) y_i(x_i)^T y_j(x_j) - \lambda_0(y_0^T y_0 - 1) \\ &- \textstyle\sum_{i,x_i} \lambda_{i,x_i}(y_i(x_i)^T y_i(x_i) - y_0^T y_i(x_i)) \\ &+ \gamma(y_0^T y_0 - 1)^2 + \gamma \textstyle\sum_{i,x_i}(y_i(x_i)^T y_i(x_i) - y_0^T y_i(x_i))^2 \end{aligned}$$

λ is the Lagrangian multiplier and γ controls the penalty term for constraint violation. To minimize $L(y, \lambda, \gamma)$, we alternately optimize y and update λ, γ. To optimize with respect to y, both objective and gradient can be efficiently computed in $O(|E|m^2 r)$, which is comparable to the complexity of a message-passing iteration in convex belief propagation. Interestingly, the gradient step could be also seen as a message passing algorithm with r-dimensional vectors for each x_i passed along edges. We use limited-memory BFGS algorithm to get a local optimal solution of y with little extra memory cost. To update λ and γ, augmented Lagrangian update is applied,

$$\begin{cases} \lambda_0 &= \lambda_0 - 2\gamma(y_0^T y_0 - 1) \\ \lambda_{i,x_i} &= \lambda_{i,x_i} - 2\gamma(y_i(x_i)^T y_i(x_i) - y_0^T y_i(x_i)) \\ \gamma &= \gamma\delta \end{cases}$$

where δ is a constant in $(0, 1)$. In most cases, the optimization converges within a few augmented Lagrangian steps. Finally, we recover the solution $b^{(2)*}$ from $y_i(x_i)$. In our experiments, we used two augmented Lagrangian steps with $\gamma \in \{50, 2000\}$ and found it works well in practice. Each Lagrangian step usually takes about 20-200 steps to reach convergence in our experiments.

It is worth noting that [5] has shown that the global optimal solution of a SDP problem can be obtained if the rank r is sufficiently large, although the transformed optimization problem is non-convex. Empirically, r can be much smaller than the bound and even a small r can produce pretty good solutions.

Theorem 1 *[5] The local optima of the low-rank method is the optimal solution of the original SDP, if $r \geq \max_{r \in \bar{N}}\{r : r(r+1) \leq 2M\}$, where M is the number of linear constraints.*

In addition to low computational complexity, another advantage of the low-rank technique is that it is a natural representation of SDP solution for sophisticated rounding schemes. Most SDP rounding schemes require Cholesky decomposition of the resultant PSD matrix, which has an high time complexity of $O(n^3m^3)$. The solution of low-rank technique can be seen as a low-rank approximation of the Cholesky decomposition and can be directly used for rounding. Empirically, the performance of low-rank representation is expected to be very close to the complete matrix representation, since the spectrum of the matrix is usually highly dominated at its several largest components.

5 Rounding Schemes for MAP Inference

In linear programming relaxation setting, MAP inference can be solved by a convex max product algorithm. Likewise, we could directly apply the convex max product algorithm to solving the first slave problem. However, simply setting ϵ to 0 will cause the first slave problem to not be strictly convex, causing convex max product to sometimes get stuck in non-optimal corners of the local polytope. To address this issue, we use a smoothed convex belief propagation algorithm by setting ϵ to a very small positive constant number. Let OBJ_ϵ denote the optimal solution of the smoothed optimization problem and OBJ_0 the solution of non-smoothed version. By applying the inequality $e^{max\{x_i\}} \leq \sum_{i=1}^n e^{x_i} \leq ne^{max\{x_i\}}$, we have $OBJ_\epsilon \leq OBJ_0 \leq OBJ_\epsilon + \epsilon m \ln n$. Setting $\epsilon = \frac{\delta}{m \ln n}$, we can get a δ-close solution to the first slave problem. After solving MAP inference via the above algorithm, we can recover all $b^*_{i,j}(x_i, x_j)$ and $b^*_i(x_i)$. If the local belief $b^*_i(x_i)$ is not integral, rounding schemes can be applied to obtain optimal or near-optimal solutions. The rounding techniques developed for SDP relaxations of MAXCUT, unique games and constraint satisfaction programs [7, 15, 16] can also be applied for MAP inference.

Naive Randomized Rounding. For each node i, we sample x_i according to the probability distribution b^*_i. It can be shown that this naive rounding method works well when the distribution is highly concentrated [2]. Probabilities of naive randomized rounding $b^*_i(s) = y_i(s)^T y_0$ could also be seen as the projected length of $y_i(s)$ to a fixed direction y_0.

Shifted Random Projection Rounding. Inspired by recent theoretical work on MAX-2SAT [16], where this more sophisticated rounding scheme was shown to be necessary for obtaining approximation guarantees, this method combines random projection rounding and naive rounding: Let w be an r-dimensional random vector with i.i.d. normally distributed components. Each x_i is assigned to state s with the largest $((1-\beta)y_0 + \beta w)^T y_i(s)$, where β is a coefficient between 0 and 1. The optimal β can be determined by enumerating a set of possible values between 0 and 1. The theoretical bound for the expected performance of spectral rounding and naive rounding [2] can be also extended to this shifted random projection rounding schemes.

6 Experiments

In this Section, we present an empirical evaluation of our method, and a comparison with other methods, both on synthetic lattice problems, and on a real-world protein side-chain packing benchmark. On the synthetic data, we evaluate both marginal probability inference and MAP inference. All the experiments are performed on a computer with a single-core AMD Opteron 2.4GHz CPU and 2G RAM.

Synthetic Lattice Problems. We generated 100 random instances of 10×10 grids with binary variables $x_i \in \{0, 1\}$. Node potentials $\theta_i(x_i)$ were drawn from $U[-0.05, 0.05]$ and edge potentials $\theta_{i,j}(x_i, x_j)$ were drawn from $U[-\kappa, \kappa]$, where parameter κ controls the coupling strength.

First, we evaluated our method on the task of marginal probability and partition function estimation, and compared its performance to that of tree-reweighted belief propagation (TRBP). Note that the recently proposed MPLP method [19] does not immediately lend itself to estimating marginal probabilities or the partition functions[1], nor does a straight-forward SDP relaxation approach.

The left panel of Figure 1 shows the L_1 error of marginal probability estimation (for all i, x_i) with respect to the coupling strength, for both TRBP and for our method. In these experiments, the damping

[1] T. Hazan recently presented in a workshop talk how MPLP might be modified for marginal probability estimation, but to the best of our knowledge this has never been implemented, verified, and experimented with.

factor for TRBP is set to 0.5 and we used a rank of 5 for the SDP updates in our method. For small κ, the problem is easy for both TRBP and our method. However, when the coupling strength increases, the errors of TRBP increase dramatically, but our method still obtains reasonably good results. We also show, in the middle panel of Figure 6, the looseness of the upper bounds on the partition function obtained in these procedures. As can be seen, tightening the local marginal constraints with the semi-definite constraint indeed yields a tighter bound on the partition function for the grid, especially with stronger edge potentials.

We now turn to MAP inference, where in addition to TRBP we also compared with a "pure" SDP approach (using only the semi-definite outer bound), solving it with the same low-rank method used by our approach. We compared the different methods' ability to obtain optimal integer assignments. The solutions of TRBP are obtained by rounding the labels with maximal local beliefs. For the pure SDP and for our combined approach, we first experimented with the naive rounding (choosing the state with the highest belief). The failure rates for the different approaches (over 100 random instances) are shown in the right panel of Figure 6. As can be seen, when the coupling potentials are weak, TRBP yields better results that the pure SDP approach. However, when κ is large (i.e. the edge potentials are strong), the semi-definite constraint is better then the local marginal constraints used by TRBP. In any case, our method, which combines the two constraints, even without any rounding, is always superior to either TRBP or the pure SDP approach.

Randomized rounding can further increase the performance of our method. We tested both the naive randomized rounding, choosing the highest likelihood of 100 randomizations, and the shifted random projection rounding schemes with 100 random rounding for each $\beta \in \{0, 0.1, 0.2, ..., 1\}$. The failure rates after using these randomized rounding schemes are also plotted on the right panel of Figure 6. As can be seen, the naive rounding decreases the failure rate significantly, and with the shifted random projection rounding, we can find the optimal integer solution for all grid instances. This result indicates that our randomized rounding schemes, especially the shifted rounding scheme, is very effective in dealing with the difficult cases on which linear programming relaxation does not work well.

We also compared our algorithm with the state-of-the-art MPLP algorithm with tightening [19] on these grid graphs. We used Sontag's implementation downloaded from his website. We ran MPLP with at most 1000 tightening steps. Both triplets and neighboring blocks are tested for tightening. This algorithm performs better than our method when the coupling strength of the graph is relatively small. But for graphs with strong edge weights, our method solves about 10% more instances than MPLP with tightening. What we see here is that tightening by adding higher order local marginal constraints, as in MPLP, and tightening by adding a semi-definite constraint can each be advantageous in a different regime. It is certainly conceivable to use a combined approach for particularly hard problems, using both higher order local marginal constraints and a semi-definite constraint, and perhaps also iteratively adding higher order constraints as in MPLP.

Finally, to show the accuracy and efficiency of low-rank method in dealing with semidefinite constraints, we compare our algorithm with a state-of-the-art interior-point solver DSDP [1]. We tested both optimization approaches on the second separated subproblem with 100 random instances, each of which is a 10×10 grid with $\kappa = 2$. We ran DSDP with tolerance $= 10^{-4}$. The averaged primal solution relative difference between DSDP and our low-rank SDP algorithm is less than 0.1%, 0.05%, 0.02% and 0.01%, when 5-rank, 10-rank, 20-rank and 50-rank are used, respectively. On average, the running time of our low-rank SDP algorithm is $11s$ for rank = 5, 15s for rank = 10, 27s for rank = 20, 62s for rank = 50 respectively, while the running time of DSDP solver is $642s$. By contrast, the running time of our TRBP implementation is 6s. For most instances, the total running time of our algorithm is less than 10 times that of TRBP.

Protein side-chain packing. The goal of protein side-chain packing is to predict the side-chain rotamer positions of each amino acid by minimizing a given energy function. For consistency with many other methods in the literature, we use Rosetta energy function for side-chain packing. The side-chain packing problem can be formulated as a MAP inference problem of Markov random fields. In [26], the authors show that TRBP can find the optimal MAP assignment for 339 out of 369 test instances. For the most challenging 30 instances, the optimal solutions of LP relaxation are fractional. We tested our algorithm on these 30 challenging instances with rank 10 in low-rank technique. We also tested our algorithm with higher rank, e.g. 20 and 30, and didn't find much difference in the resulting assignments.

The primal objective values achieved by our algorithm are better than or equivalent to TRBP for all the 30 instances. Meanwhile, our algorithm can solve 26 instances to optimal without any rounding. The primal objective values of the other 4 instances are within a gap of 0.5% of the optimal energy. In contrast, the averaged gap of tree-reweighted belief propagation is

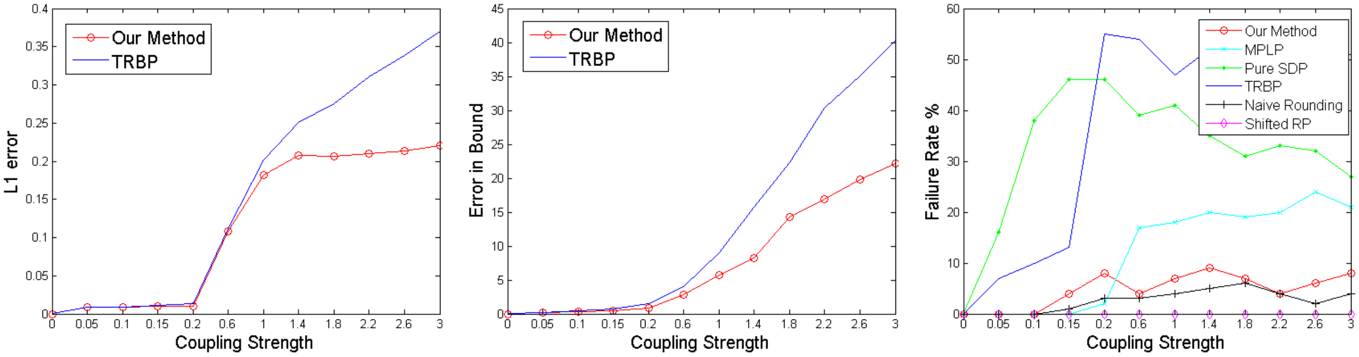

Figure 1: **Experiments on a 10x10 grid. Left: average L_1 errors of TRBP and our algorithm for marginal probability estimation. Middle: average looseness of the resulting upper bound on the log partition function. Right: failure rates of MAP inference for TRBP, pure SDP, MPLP and our algorithm without and with the different rounding schemes. A method is considered "successful" on an instance if it yields the optimal integer solution.**

Table 1: **Averaged performance of low-rank SDP on the second SDP subproblem on 100 random grid graphs. The subproblems are generated after the first decomposition step.**

Method	DSDP	LRSDP-5	LRSDP-10	LRSDP-20	LRSDP-50
Running Time	642s	11s	15s	27s	62s
Relative Errors(%)	0	0.1	0.05	0.02	0.01

3.0% and the maximal gap is 10.2%. By generating 100 random samples with shifted rounding schemes for each $\beta \in [0, 0.1, 0.2, ..., 1.0]$, our algorithm can solve all the 30 instances to optimal. We also observed that 15 of the 30 instances can be solved to optimal within 6 iterations of updating the dual variable v in the master problem.

It has been reported in [19] that an adaptive-augmented Kikuchi method (MPLP with tightening) can also solve all the 369 protein side-chain packing instances. This method uses a linear programming relaxation with triplets to gradually tighten local polytope into a partial Kikuchi polytope and then employs a message passing algorithm to find the MAP assignment. The running times of the MPLP algorithm are between 1 minute and 1 hour with 9 minutes as the median. Our method has running times between 3 minutes and 1.8 hours with 13 minutes as the median. The low-rank SDP step alone costs 34% of the overall running time. Our code is implemented with Matlab, we expect that the running time of our algorithm will be significantly reduced if implemented with C/C++. A pure SDP algorithm has also been proposed for protein side-chain packing [2]. However, this method uses only a semidefinite constraint but not the local polytope and works for only very small proteins. No large-scale experiments are conducted to test this SDP algorithm.

7 Conclusion

We proposed a new algorithm for two inference tasks on graphical models. Our approach obtains a tighter outer bound on the marginal polytope by intersecting two relaxations: the local polytope and the semidefinte outer bound. With the dual decomposition framework, we separated the optimization problem into two slave problems, a SDP solved by a low-rank SDP algorithm, and a free-energy functional optimization problem solved by convex belief propagation. Our algorithm shows superior performance over tree-reweighted belief propagation on both synthetic data and a protein side-chain packing benchmark. Our method is competitive with the state-of-the-art MPLP method, solving all protein tasks solved by MPLP, and beating MPLP on latices with strong edge potentials. The lattice experiments demonstrate that our proposed method and MPLP are incomparable and are each advantageous in different regimes, suggesting a combined approach could be particularly advantageous. In future work we expect to apply dual decomposition methods to high-order graphs, mainly the Kikuchi polytope for tightening the marginalization constraints as well as the Lasserre hierarchy for tightening SDP relaxations. In addition, other rounding schemes will also be investigated for MAP inference, e.g. SDP rounding schemes based on Grothendieck's inequality for integer quadratic programming.

References

[1] http://www.mcs.anl.gov/hs/software/dsdp/.

[2] M. Singh B. Chazelle, C. Kingsford. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS Journal on Computing*, 2004.

[3] Inderjit Dhillon Brian Kulis, Suvrit Sra. Convex perturbations for scalable semidefinite programming. In *AISTATS*, 2009.

[4] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 2003.

[5] S. Burer and R.D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 2005.

[6] A. Globerson and T. Jaakkola. Fixing max-product: convergent message passing algorithms for MAP relaxations. 2007.

[7] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 1995.

[8] T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Conference on Uncertainty in Artifical Intelligence (UAI)*, Helsinki, Finland, July 2008.

[9] T. Heskes. Convexity Arguments for Efficient Minimization of the Bethe and Kikuchi Free Energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006.

[10] Sameer Sheorey Kaushik Mitra and Rama Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[11] M. Pawan Kumar, V. Kolmogorov, and P. Torr. An analysis of convex relaxations for map estimation of discrete mrfs. *Journal of Machine Learning Research (JMLR)*, 2009.

[12] Jason Lee, Ben Recht, Ruslan Salakhutdinov, Nathan Srebro, and Joel A. Tropp. Practical large-scale optimization for max-norm regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[13] P. Torr M. Pawan Kumar and A. Zisserman. Solving Markov Random Fields using Second Order Cone Programming Relaxations. In *In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] O. Meshi, Ariel Jaimovich, A. Globerson, and N. Friedman. Convexifying the bethe free energy. In *Conference on Uncertainty in Artifical Intelligence (UAI)*, 2009.

[15] Yury Makarychev Moses Charikar, Konstantin Makarychev. Near-optimal algorithms for unique games. In *STOC*, 2006.

[16] Yury Makarychev Moses Charikar, Konstantin Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *SODA*, 2007.

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

[18] P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *In International Conference on Machine learning (ICML) 23*, pages 737–744, 2006.

[19] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening lp relaxations for map using message passing. In *In Uncertainty in Artificial Intelligence (UAI)*, 2008.

[20] David Sontag, Amir Globerson, and Tommi Jaakkola. Introduction to dual decomposition for inference. 2011.

[21] P.H.S. Torr. Solving markov random fields using semi definite programming. In *International Workshop on Artificial Intelligence and Statistics (AISTAT)*, 2003.

[22] M. J. Wainwright and M. I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. In *IEEE Transactions on Signal Processing, Vol. 54(6), pages 2099–2109*, 2006.

[23] M. J. Wainwright and M. I. Jordan. *Graphical models, exponential families, and variational inference*. 2008.

[24] MJ Wainwright, TS Jaakkola, and AS Willsky. A new class of upper bounds on the log partition function. *Information Theory, IEEE Transactions on*, 51(7):2313–2335, 2005.

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

[26] C. Yanover, T. Meltzer, and Y. Weiss. Linear Programming Relaxations and Belief Propagation–An Empirical Study. *The Journal of Machine Learning Research*, 7, 2006.