On Iteratively Constraining the Marginal Polytope for Approximate Inference and MAP

David Sontag
CSAIL
Massachusetts Institute of Technology
Cambridge, MA 02139

Tommi Jaakkola
CSAIL
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We propose a cutting-plane style algorithm for finding the maximum a posteriori (MAP) state and approximately inferring marginal probabilities in discrete Markov Random Fields (MRFs). The variational formulation of both problems consists of an optimization over the marginal polytope, with the latter having an additional non-linear entropy term in the objective. While there has been significant progress toward approximating the entropy term, the marginal polytope is generally approximated by the local consistency constraints, which give only a loose outer bound. Our algorithm efficiently finds linear constraints that are violated by points outside of the marginal polytope, making use of the cut polytope, which has been studied extensively in the context of MAX-CUT. We demonstrate empirically that our algorithm finds the MAP solution for a larger class of MRFs than before. We also show that tighter yet efficient relaxations of the marginal polytope result in more accurate pseudomarginals.

1 INTRODUCTION

Markov Random Fields (MRFs) have been useful across a wide spectrum of problems, from computer vision and natural language processing to computational biology. The utility of such models depends critically on fast and accurate inference calculations, typically either finding the most likely setting of all the variables (referred to here as the MAP problem) or evaluating marginal probabilities of specific subsets of the variables. With the exception of low tree-width MRFs, or specific subclasses of models such as restricted planar graphs, solving either of these inference problems requires approximate methods.

We will consider here a subclass of MRFs, those that are naturally expressed in terms of pairwise dependencies (potentials) between the variables[1]. In this context, the challenging nature of inference calculations can be traced back to the difficulty of working with what is known as the *marginal polytope*. This is the set of marginal probabilities arising from valid MRFs with the same structure, i.e., marginal probabilities that are *realizable*. In general, unless P=NP, it is not possible to give a polynomial number of linear constraints characterizing the marginal polytope (a point we will make precise in Appendix B). However, for particular classes of graphs, such as trees and planar graphs, a small number of constraints indeed suffice to fully characterize the marginal polytope.

Finding the MAP assignment for MRFs with pairwise potentials can be cast as an integer linear program over the marginal polytope. This problem is also known as the MAX-CUT problem and has been extensively studied in the mathematical programming literature. The approximate methods solve the linear program over an easier to handle outer bound on the marginal polytope. For example, the simple linear programming relaxation of the MAP problem corresponds to optimizing over the *local marginal polytope* characterized by pairwise consistent marginals. These simpler linear programs can be often solved efficiently via message passing algorithms (e.g., tree-reweighted max-product described in [9]). The famous Goemans and Williamson's method has the best known approximation ratio and uses an outer bound based on semidefinite constraints, together with randomized rounding. Another approach, often used within combinatorial optimization, are cutting-plane algorithms. They find linear inequalities that separate the current frac-

[1]MRFs with higher order interactions can in principle be mapped back to MRFs with pairwise potentials though with a possible loss in representational power or efficiency of inference calculations.

tional solution from all feasible integral solutions, iteratively adding such constraints into the linear program and thereby adaptively tightening the approximation of the marginal polytope.

The marginal polytope also plays a critical role in approximate (or exact) variational evaluation of marginal probabilities in MRFs (cf. convex duality and the exponential family of distributions [12]). For example, in the tree-reweighted sum-product (TRW) algorithm of Wainwright et al. [10] the inference problem is posed as a convex optimization problem over the marginal polytope, then relaxed to an outer bound. The additional difficulty in this context is in representing the entropy corresponding to any approximate (pseudo) marginal probabilities. The relative value of the entropy approximation in comparison to the relaxation of the marginal polytope is not well-understood. We will address this point further in the paper.

The main contribution of our work is to show how to achieve tighter outer bounds on the marginal polytope in an efficient manner using the cutting-plane methodology, iterating between solving a relaxed problem and adding additional constraints. While extensively used for solving integer linear programs, such methods have yet to be demonstrated in the context of evaluating marginals. One key intuition for why this type of algorithm may be successful is that the marginal polytope only needs to be well-specified near the optimum of the objective, and that for real-world problems that have structure, only a small number of constraints may be necessary to sufficiently constrain the marginal polytope at the optimum. The approach can also be used as an anytime algorithm, allowing us to trade-off increased running time for possibly better approximations.

1.1 RELATED WORK

A number of existing methods for evaluating marginals can be related to approximations of the marginal polytope. Mean field methods, for example, use *inner bounds* on the marginal polytope by restricting the approximating distributions to lie within specific families of distributions, subsets of the model in question. As a result, one obtains a *lower bound* on the log-partition function as opposed to an upper bound characteristic of outer bound approximations. The advantage of inner bounds is that for these points within the marginal polytope, e.g., trees or completely factored distributions, the corresponding entropy functions have closed form expressions. The primary disadvantage is the loss of convexity and the accompanying difficulties with locally optimal solutions.

Most message passing algorithms for evaluating marginals, including belief propagation (sum product) and tree-reweighted sum-product (TRW), operate within the local marginal polytope. In TRW, for example, the key contribution involves the entropy function rather than the marginal polytope. The entropy is decomposed into a weighted combination of entropies of tree-structured distributions with the same pairwise marginals.

Stronger effective constraints on the marginal polytope can be obtained by decomposing the model in terms of planar graphs as opposed to trees (Globerson et al. [8]). The marginal polytope for a class of planar graphs can be fully characterized using so called *triangle inequalities* (see below). A different type of restriction on the marginal polytope comes from semi-definite constraints (any valid covariance matrix has to be positive semi-definite). Such a restriction on the marginal polytope can be enforced either explicitly in the context of MAP, or implicitly through a log-determinant approximation to the entropy when evaluating marginals. The log-determinant serves as a barrier function for selecting approximate marginals (Wainwright and Jordan [11]).

Previous work most related to ours is by Barahona et al. [2] in the context of finding the MAP assignment in Ising models. Their approach iterates between solving the LP and adding in constraints corresponding to violated *cycle inequalities* (discussed below). Our key observation is that similar ideas can be used to approximately solve any objective function which is defined on the marginal polytope $\mathcal{M}$. In particular, we can use any approximation of the entropy (e.g. TRW or log-determinant) to find pseudomarginals.

2 BACKGROUND

2.1 MARKOV RANDOM FIELDS

We consider Markov Random Fields (MRFs) with pairwise potentials. Given a graph $G = (V, E)$ with vertices V and edges E, the model is parameterized by potential functions defined on the edges $(i, j) \in E$ in the graph. To simplify the exposition we will restrict ourselves to the case of binary variables, $X_i \in \{0, 1\}$, and provide the multinomial extension in Appendix A. The joint distribution over $X = \{X_1, \ldots, X_n\}$ is now given by:

$$\begin{aligned} \log P(X;\vec{\theta}) &= \sum_{i\in V} \theta_i X_i + \sum_{(i,j)\in E} \theta_{ij} X_i X_j - A(\vec{\theta}) \\ &= \langle \vec{\theta}, \vec{\phi}(X) \rangle - A(\vec{\theta}) \end{aligned} \quad (1)$$

where $A(\vec{\theta})$ is the log-normalization (partition) function and the vector $\vec{\phi}(X)$ of dimension $d = |V| + |E|$

collects together X_i for $i \in V$ and X_iX_j for $(i,j) \in E$. The log-partition function plays a critical part in the inference calculations.

The inference task is to evaluate the mean vector $\vec{\mu} = E_\theta[\vec{\phi}(X)]$ containing the sufficient statistics $\mu_i = E_\theta[X_i]$ for $i \in V$, and $\mu_{ij} = E_\theta[X_iX_j]$ for $(i,j) \in E$. Knowing $A(\vec{\theta})$ would suffice to calculate $\vec{\mu}$ since,

1. $\frac{\partial A(\vec{\theta})}{\partial \theta_i} = E_\theta[X_i] = \mu_i$,
2. $\frac{\partial^2 A(\vec{\theta})}{\partial \theta_i \partial \theta_j} = E_\theta[X_iX_j] - E_\theta[X_i]E_\theta[X_j] = \mu_{ij} - \mu_i\mu_j$

$A(\vec{\theta})$ is clearly convex in the parameters $\vec{\theta}$ since the second moment matrix is positive semi-definite. This suggests an alternative definition of the log-partition function, in terms of its Fenchel-Legendre conjugate [12]

$$A(\vec{\theta}) = \sup_{\vec{\mu} \in \mathcal{M}} \left\{ \langle \vec{\theta}, \vec{\mu} \rangle - B(\vec{\mu}) \right\}, \tag{2}$$

where $B(\vec{\mu}) = -H(\vec{\mu})$ is the negative entropy of the distribution parameterized by $\vec{\mu}$ and is also convex. $\mathcal{M}$ is the set of realizable mean vectors $\vec{\mu}$ known as the *marginal polytope*[2]:

$$\mathcal{M} := \left\{ \vec{\mu} \in \mathbb{R}^d \mid \exists p(X) \text{ s.t. } \begin{array}{l} \mu_i = E_p[X_i], \\ \mu_{ij} = E_p[X_iX_j] \end{array} \right\} \tag{3}$$

The value $\vec{\mu}^* \in \mathcal{M}$ that maximizes (2) is precisely the desired mean vector corresponding to $\vec{\theta}$. In general both $\mathcal{M}$ and the entropy $H(\vec{\mu})$ are difficult to characterize. We can try to obtain the mean vector approximately by using an outer bound on the marginal polytope and by bounding the entropy function. We will demonstrate later in the paper that tighter outer bounds on $\mathcal{M}$ are valuable, especially for realistic models where the couplings θ_{ij} are large.

The MAP problem is to find the assignment $X = x$ which maximizes $P(x; \vec{\theta})$, or equivalently

$$\max_{x \in \{0,1\}^n} \log P(x; \vec{\theta}) = \max_{x \in \{0,1\}^n} \langle \vec{\theta}, \vec{\phi}(x) \rangle - A(\vec{\theta}) \tag{4}$$

$$= \sup_{\vec{\mu} \in \mathcal{M}} \langle \vec{\theta}, \vec{\mu} \rangle - A(\vec{\theta}) \tag{5}$$

where the log-partition function $A(\vec{\theta})$ is a constant for the purpose of finding the maximizing assignment and can be ignored. The last equality comes from the fact that the distribution whose mean vector attains the maximum is simply the one peaked at the maximizing assignment x^*. In other words, the maximizing $\vec{\mu}^* = \vec{\phi}(x^*)$ (if unique). In summary, both inferring marginals and the MAP assignments correspond to optimizing some objective over the marginal polytope $\mathcal{M}$.

[2]The definition here is adapted to the case of binary variables.

2.2 THE CUT POLYTOPE

In this section we will show that the marginal polytope[3] is equivalent to the cut polytope, which has been studied extensively within the fields of combinatorial and polyhedral optimization [3, 1, 7]. This equivalence enables us to translate relaxations of the cut polytope into relaxations of the marginal polytope.

Definition 1. Given a graph $G = (V, E)$ and $S \subseteq V$, let $\delta(S)$ denote the vector of $\mathbb{R}^E$ defined for $(i,j) \in E$ by,

$$\delta(S)_{ij} = 1 \text{ if } |S \cap \{i,j\}| = 1, \text{ and 0 otherwise.} \tag{6}$$

In other words, the set S gives the cut in G which separates the nodes in S from the nodes in $V \setminus S$; $\delta(S)_{ij} = 1$ when i and j have different assignments. The *cut polytope* projected onto G is the convex hull of the above cut vectors:

$$\text{CUT}^\square(G) = \Big\{ \sum_{S \subseteq V_n} \lambda_S \delta(S) \mid \sum_{S \subseteq V_n} \lambda_S = 1 \text{ and} \tag{7}$$

$$\lambda_S \geq 0 \text{ for all } S \subseteq V_n \Big\}. \tag{8}$$

The cut polytope for the complete graph on n nodes is denoted simply by $\text{CUT}^\square_n$. We should note that the cut cone is of great interest in metric embeddings, one of the reasons being that it completely characterizes ℓ_1-embeddable metrics [7].

2.2.1 Equivalence to Marginal Polytope

Suppose that we are given a MRF defined on the graph $G = (V, E)$. To give the mapping between the cut polytope and the marginal polytope we need to construct the *suspension graph* of G, denoted ∇G. Let $\nabla G = (V', E')$, where $V' = V \cup \{n+1\}$ and $E' = E \cup \{(i, n+1) \mid i \in V\}$. The suspension graph is necessary because a cut vector $\delta(S)$ does not uniquely define an assignment to the vertices in G – the vertices in S could be assigned either 0 or 1. Adding the extra node allows us to remove this symmetry.

Definition 2. The linear bijection ξ from $\mu \in \mathcal{M}$ to $\vec{x} \in \text{CUT}^\square(\nabla G)$ is given by $x_{i,n+1} = \mu_i$ for $i \in V$ and $x_{ij} = \mu_i + \mu_j - 2\mu_{ij}$ for $(i,j) \in E$.

Using this bijection, we can reformulate the MAP problem from (5) as a MAX-CUT problem:

$$\sup_{\vec{\mu} \in \mathcal{M}} \langle \vec{\theta}, \vec{\mu} \rangle = \max_{x \in \text{CUT}^\square(\nabla G)} \langle \vec{\theta}, \xi^{-1}(x) \rangle. \tag{9}$$

Furthermore, any valid inequality for the cut polytope can be transformed into a valid inequality for the

[3]In the literature on cuts and metrics (e.g. [7]), the marginal polytope is called the *correlation polytope*, and is denoted by $\text{COR}^\square_n$.

marginal polytope by using this mapping. In the following sections we will describe several known relaxations of the cut polytope, all of which directly apply to the marginal polytope by using the mapping.

2.2.2 Relaxations of the Cut Polytope

It is easy to verify that every cut vector $\delta(S)$ (given in equation 6) must satisfy the triangle inequalities: $\forall i, j, k$,

$$\begin{aligned} \delta(S)_{ik} + \delta(S)_{kj} - \delta(S)_{ij} &\geq 0 \\ \delta(S)_{ij} + \delta(S)_{ik} + \delta(S)_{jk} &\leq 2. \end{aligned}$$

Since the cut polytope is the convex combination of cut vectors, every point $x \in \mathrm{CUT}_n^{\square}$ must also satisfy the triangle inequalities. The *semimetric polytope* $\mathrm{MET}_n^{\square}$ consists of those points $x \geq 0$ which satisfy the triangle inequalities. Note that the projection of these $O(n^3)$ inequalities onto an incomplete graph is non-trivial and will be addressed in the next section. If we instead consider only those constraints that are defined on the vertex $n+1$, we get a further relaxation, the *rooted semimetric polytope* $\mathrm{RMET}_n^{\square}$.

We could now apply the inverse mapping ξ^{-1} to obtain the corresponding relaxations for the marginal polytope. The $\xi^{-1}(\mathrm{RMET}_n^{\square})$ polytope is the same as the local marginal polytope LOCAL, which fully characterizes the marginal polytope of any tree-structured distribution:

$$\mathrm{LOCAL} := \left\{ \vec{\mu} \in \mathbb{R}_+^d \mid \begin{array}{l} \forall (i,j) \in E \\ \mu_{ij} \leq \mu_{ii}, \mu_{ij} \leq \mu_{jj} \\ \mu_{ii} + \mu_{jj} - \mu_{ij} \leq 1 \end{array} \right\} \quad (10)$$

Interestingly, the triangle inequalities suffice to describe $\mathcal{M}$ (i.e. $\mathcal{M} = \xi(\mathrm{MET}^{\square}(\nabla G))$) for a graph G if and only if G has no K_4-minor[4]. It can be shown that both LOCAL and $\mathcal{M}$ have the same integral vertices [12, 7], which is one of the reasons why the LOCAL polytope provides a natural relaxation for MAP.

3 CUTTING-PLANE ALGORITHM

The main result in this paper is the proposed algorithm given in Table 1. The algorithm iterates between solving for an upper bound of the log-partition function (see eqn. (2)) and tightening the outer bound on the marginal polytope by adding constraints that are violated by the pseudomarginals at the optimum μ^*. Any approximation $B^*(\vec{\mu})$ of the entropy function can be used with our algorithm, as long as we can efficiently do the optimization given in line 3. In particular, we have investigated using the log-determinant relaxation [11] and the TRW relaxation [10]. They have two particularly appealing features. First, both give upper bounds on the entropy function, and thus allow our algorithm to be used to give tighter upper bounds on the log-partition function[5]. Second, the resulting objectives are convex, allowing for efficient optimization using conditional gradient or other methods. The algorithm for MAP is the same, but excludes the entropy function in line 3; the optimization is simply a linear program.

Table 1: Inference Algorithm for Psuedomarginals

1.	**(initialize)** $\mathcal{R} \leftarrow$ **LOCAL.**
2.	**Loop:**
3.	Solve optimization $\max_{\vec{\mu} \in \mathcal{R}} \left\{ \langle \vec{\theta}, \vec{\mu} \rangle - B^*(\vec{\mu}) \right\}$.
4.	Construct ∇G and assign weights $w = \xi(\mu^*)$.
5.	Run separation algorithms from Table 2.
6.	Add violated inequalities to $\mathcal{R}$. If none, stop.

We begin with the loose outer bound on the marginal polytope given by the local consistency constraints. It is also possible to use a tighter initial outer bound. For example, we could include the constraint that the second moment matrix is positive semi-definite, as described by Wainwright and Jordan [11]. The disadvantage is that it would require explicitly representing all $O(n^2)$ μ_{ij} variables[6], which may be inefficient for large yet sparse MRFs.

3.1 SEPARATION ALGORITHMS

Here we list some of the separation algorithms that are known for the cut polytope. Each algorithm separates a different class of inequalities. All of these inequalities arise from the study of the facets[7] of the cut polytope. The triangle inequalities, for example, are a special case of a more general class of inequalities called the hypermetric inequalities [7] for which efficient separation algorithms are not known. Another class, the Clique-Web inequalities, contains three spe-

[4]This result is applicable to any binary pairwise MRF. However, if we are given an Ising model without a field, then we can construct a mapping to the cut polytope without using the suspension graph. By the corresponding theorem in [7], CUT(G)=MET(G) when the graph has no K_5 minor, so it would be exact for planar Ising models with no field.

[5]In principal, our algorithm could be used with any approximation of the entropy function, e.g. the Bethe free energy approximation, which would not lead to an upper bound on the log partition function, but may provide better pseudomarginals.

[6]For triangulated graphs, it suffices to constrain the maximal cliques to be PSD.

[7]A *facet* is a polygon whose corners are vertices of the polytope, i.e. a maximal (under inclusion) face.

cial cases for which efficient separation are known, the cycle inequalities, odd-wheel and bicycle odd-wheel inequalities.

3.1.1 Cycle Inequalities

To directly optimize over the semimetric polytope $\text{MET}_n^{\square}$ we would need to represent $O(n^2)$ edge variables and $O(n^3)$ triangle inequalities, even if the graph itself was sparse (e.g. a grid Ising model). This substantial increase in complexity is perhaps the main reason why they have not been used thus far for approximate inference.

The cycle inequalities are a generalization of the triangle inequalities. They arise from the observation that any cycle in a graph must be cut an even (possibly zero) number of times by the graph cut. Namely, the cut must enter the cycle and leave the cycle (each time cutting one edge), and this could occur more than once, each time contributing two cut edges. The following result, due to Barahona [1], shows that the projected $\text{MET}_n^{\square}$ polytope can be defined in terms of cycle inequalities on just those edges in $G = (V, E)$:

$$\text{MET}^{\square}(G) = \left\{ \vec{x} \in \mathbb{R}_+^E \mid \begin{array}{l} x_{ij} \leq 1, \forall C \text{ cycle in } G \\ \text{and } F \subseteq C, |F| \text{ odd}, \\ x(F) - x(C \backslash F) \leq |F| - 1 \end{array} \right\}$$

where C is a set of edges forming a cycle in G and $x(F) = \sum_{(i,j) \in F} x_{ij}$. Furthermore, the cycle inequality for a chordless circuit C defines a facet of the $\text{CUT}^{\square}(G)$ polytope [3].

In general there are exponentially many cycles and cycle inequalities for a graph G. However, Barahona and Mahjoub [3, 7] give a simple algorithm to separate the whole class of cycle inequalities. Each cycle inequality (for cycle C and any $F \subseteq C$, $|F|$ odd) can be written as:

$$\sum_{e \in C \backslash F} x_e + \sum_{e \in F} (1 - x_e) \geq 1. \tag{11}$$

To see whether a cycle inequality is violated, construct the undirected graph $G' = (V', E')$ where V' contains nodes i' and i'' for each $i \in V$, and for each $(i, j) \in E$, the edges in E' are: (i', j') and (i'', j'') with weight x_{ij}, and (i', j'') and (i'', j') with weight $1 - x_{ij}$. Then, for each node $i \in V$ we find the shortest path in G' from i' to i''. The shortest of all these paths will not use both copies of any node j (otherwise the path j' to j'' would be shorter), and so defines a cycle in G and gives the minimum value of $\sum_{e \in C \backslash F} x_e + \sum_{e \in F} (1 - x_e)$. If this value is less than 1 then we have found a violated cycle inequality; otherwise, $\vec{x}$ satisfies all cycle inequalities. Using Dijkstra's shortest paths algorithm with a Fibonacci heap [6], the separation problem can be solved in time $O(n^2 \log n + n|E|)$.

Table 2: Summary of Separation Oracle Algorithms

SEPARATION OF	COMPLEXITY
Cycle inequalities	$\text{O}(n^2 \log n + n\|E\|)$
Odd-wheel	$\text{O}(n^4 \log n + n^3\|E\|)$
Negative-type	$O(n^3)$

3.1.2 Odd-wheel Inequalities

The odd-wheel and bicycle odd-wheel inequalities [7] give a constraint that any odd length cycle C must satisfy with respect to any two nodes u, v that are not part of C:

$$x_{uv} + \sum_{e \in C} x_e - \sum_{i \in V_C} (x_{iu} + x_{iv}) \leq 0 \tag{12}$$

$$x_{uv} + \sum_{e \in C} x_e + \sum_{i \in V_C} (x_{iu} + x_{iv}) \leq 2|V_C| \tag{13}$$

where V_C refers to the vertices of cycle C. We give a sketch of the separation algorithm for the first inequality (see [7] pgs. 481-482). The algorithm assumes that the cycle inequalities are already satisfied. For each pair of nodes u, v, a new graph G' is constructed on $V \backslash \{u, v\}$ with edge weights $y_{ij} = -x_{ij} + \frac{1}{2}(x_{iu} + x_{iv} + x_{ju} + x_{jv})$. Since we assumed that all the triangle inequalities were satisfied, y must be non-negative. Then, any odd cycle C in G' satisfies (12) if and only if $\sum_{ij \in E(C)} y_{ij} \geq x_{uv}$. The problem thus reduces to finding an odd cycle in G' of minimum weight. This can be solved in time $O(n^2 \log n + n|E|)$ using an algorithm similar to the one we showed for cycle inequalities.

3.1.3 Other Separation Algorithms

Another class of inequalities for the cut polytope are the negative-type inequalities [7], which are the same as the positive semi-definite constraints on the second moment matrix [11]. While these inequalities are not facet-defining for the cut polytope, they do provide a tighter outer bound than the local marginal polytope, and lead to an approximation algorithm for MAX-CUT. If a matrix A is not positive semi-definite, a vector x can be found in $\text{O}(n^3)$ time such that $x^T A x < 0$, giving us a linear constraint on A which is violated by the current solution. Thus, these inequalities can also be used in our iterative algorithm, although the utility of doing so has not yet been determined.

If solving the relaxed problem results in a fractional solution which is outside of the marginal polytope, Gomory cuts [4] provide a way of giving, in closed form, a hyperplane which separates the fractional solution from all integral solutions. These inequalities

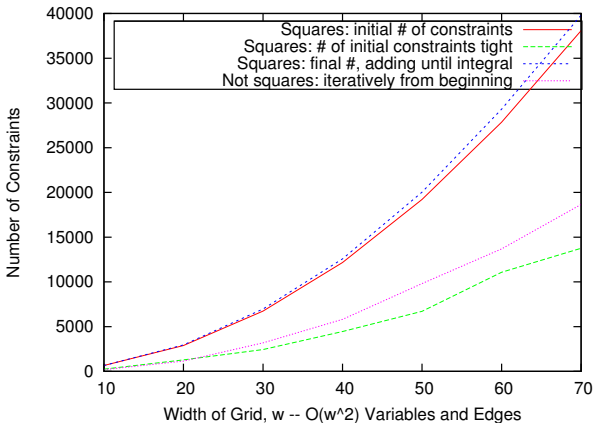

Figure 1: MAP on Ising Grid Graph.

are applicable to MAP because any fractional solution must lie outside of the marginal polytope. We show in Appendix B that it is NP-hard to test whether an arbitrary point lies within the marginal polytope. Thus, Gomory cuts are not likely to be of much use for marginals.

4 EXPERIMENTS

We experimented with the algorithm shown in Table 1 for both MAP and marginals. We used the glpkmex and YALMIP optimization packages within Matlab, and wrote the separation algorithms in Java. We made no attempt to optimize our code and thus omit running times. However, MAP for the largest MRFs finished in less than a day, and everything else was significantly faster.

4.1 MAP

Our goal in doing experiments for MAP is to demonstrate that our proposed algorithm can scale to large problems, and to show that by using our algorithm we can find the MAP solution more often than when using the LOCAL polytope relaxation. We should note that we are primarily interested in the setting where we have a certificate of optimality, which our algorithm can verify by checking that its solution is integral. Neither the max-product algorithm nor the Goemans-Williamson approximation algorithm give any such guarantee of optimality.

In Figure 1 we show results for MAP on Ising grid graphs. For each width, we generated 3 random graphs and averaged the results. The parameters were sampled $\theta_i \sim \mathcal{N}(0, .01)$ and $\theta_{ij} \sim \mathcal{N}(0, 1)$. The local consistency constraints alone were insufficient, giving fractional solutions for all trials. However, using our algorithm together with the cycle inequalities we were able to find the MAP solution for all trials. On the largest examples (70x70 grids), integral solutions are found with fewer than 20,000 constraints (see "Not squares" in figure). To contrast, note that if we had used all of the triangle inequalities directly, we would have needed over 50 billion constraints and 12 million variables. We also looked at the length of the cycles for which cycle inequalities were added. For the 50x50 grid, only 13% of the cycles were length 4, and there was a very long tail (1% of the cycles were of length 52). Thus, the cycle inequalities appear to be capturing an interesting global constraint.

Drawing insight from the success of generalized belief propagation on Ising grids, we tried initializing $\mathcal{R}$ to LOCAL plus the $O(n)$ length 4 cycle inequalities corresponding to the squares of the grid. Interestingly, we only had to add a small number of additional cycle inequalities before reaching the MAP solution, resulting in much faster running times. For structured problems such as grids, using our algorithm in this way, with a good "basis" of cycles, may be of great practical value.

While using the cycle inequalities allowed us to find the MAP solution for all of the grid models, we do not expect the same to hold for less structured MRFs. For such cases, one could try using our algorithm together with branch-and-bound (these are called branch-and-cut algorithms). We investigated whether using the separation oracle for bicycle odd-wheel inequalities was helpful for 30 and 40 node complete graphs, parameterized as before. Below 30 nodes the cycle inequalities are sufficient to find the MAP solution. We found that, in the majority of the cases where there was a fractional solution using just the cycle inequalities, the odd-wheel inequalities result in an integral solution, adding between 500 and 1000 additional constraints.

4.2 MARGINALS

In this section we show that using our algorithm to optimize over the $\xi^{-1}(\text{MET}_n^{\square})$ polytope results in significantly more accurate pseudomarginals than can be obtained by optimizing over LOCAL. We experiment with both the log-determinant [11] and the TRW [10] approximations of the entropy function. Although TRW can efficiently optimize over the spanning tree polytope, for these experiments we simply use a weighted distribution over spanning trees, where each tree's weight is the sum of the absolute value of its edge weights. The edge appearance probabilities corresponding to this distribution can be efficiently computed using the Matrix Tree Theorem [13]. We optimize the TRW objective using conditional gradient, using linear programming at each iteration to do the projection onto $\mathcal{R}$.

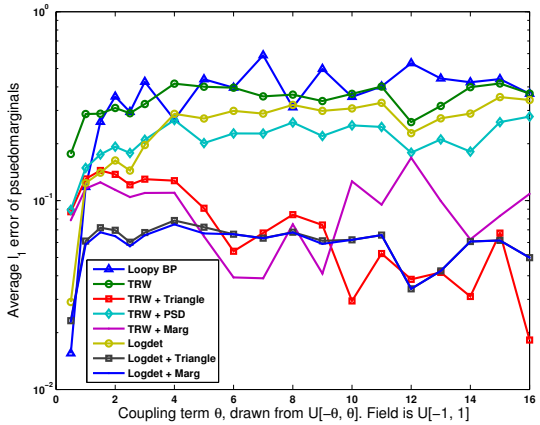

Figure 2: Accuracy of Pseudomarginals on 10 node Complete Graph.

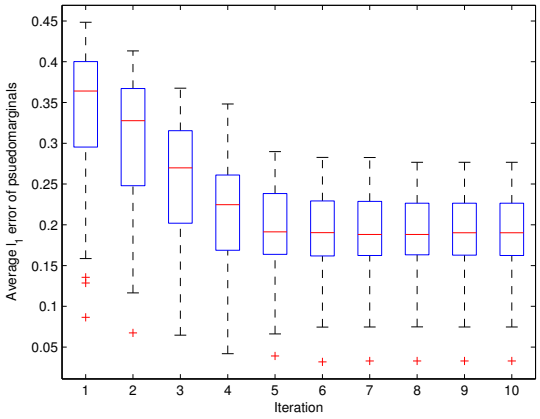

Figure 3: Convergence on 10x10 Grid with $\theta_i \in U[-1,1]$ and $\theta_{ij} \in U[-4,4]$ (40 trials).

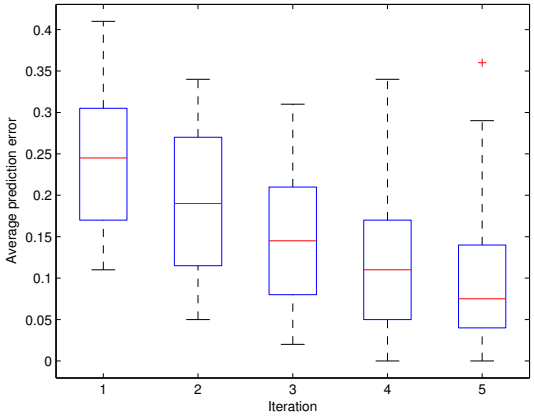

Figure 4: Convergence on 10x10 Grid with $\theta_i \in U[-1,1]$ and $\theta_{ij} \in U[-4,4]$ (40 trials).

These trials were on pairwise MRFs with $X_i \in \{-1,1\}$ (see eqn. (1)) and mixed potentials. In Figure 2 we show results for 10 node complete graphs with $\theta_i \sim U[-1,1]$ and $\theta_{ij} \sim U[-\theta,\theta]$, where θ is the coupling strength shown in the figure. Note that these MRFs are among the most difficult to do inference in, due to their being so highly coupled. For each data point we averaged the results over 10 trials. The Y-axis, given on a log-scale, shows the average ℓ_1 error of the singleton marginals. Note that although the coupling is so large, the external field is also significant, and the actual probabilities are interesting, away from .5 and not all the same (as you would find in a highly coupled model with attractive potentials).

In this difficult setting, loopy belief propagation (with a .5 decay rate) seldom converges. The TRW and log-determinant algorithms, which optimize over the local consistency polytope, give pseudomarginals only slightly better than loopy BP. Even adding the positive semi-definite constraint on the second moments, for which TRW must be optimized using conditional gradient and semi-definite programming for the projection step, does not improve the accuracy by much. However, both entropy approximations give significantly better pseudomarginals when used by our algorithm together with the cycle inequalities (see "TRW + Triangle" and "Logdet + Triangle" in the figure).

We were also interested in investigating the extent to which further tightening of the marginal polytope relaxations would improve pseudomarginal accuracy. The marginal polytope has 2^N vertices, where N is the number of variables in the binary MRF. Thus, for these small MRFs we can exactly represent the marginal polytope as the convex hull of its vertices. We show in Figure 2 the results for optimizing the TRW and log-determinant objectives over the exact marginal polytope (see "TRW + Marg" and "Logdet + Marg"). For both entropy approximations, optimizing over the $\xi^{-1}(\text{MET}_n^{\square})$ relaxation gives nearly as good accuracy as with the exact marginal polytope, and even better in some situations (this is a surprising result). Thus, for these entropy approximations, our algorithm may give as good accuracy as can be hoped for. However, these results are highly dependent on what entropy approximation is used. For example, for some MRFs, the solution to the log-determinant objective already lies within the marginal polytope (possibly because of the implicit positive semi-definite constraint given by the log barrier) although the pseudomarginals are not very accurate.

Next, we looked at the number of iterations (in terms of the loop in Table 1) the algorithm takes before all

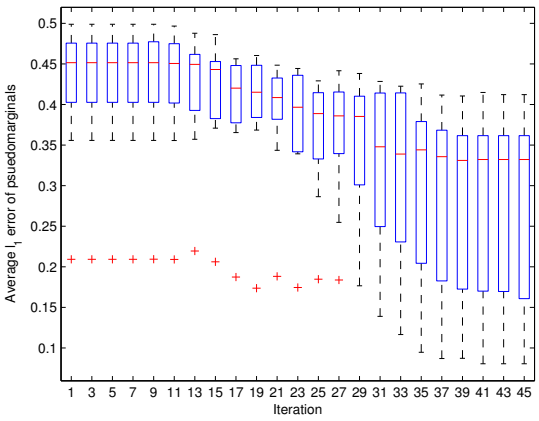

Figure 5: Convergence on 20 node Complete Graph with $\theta_i \in U[-1,1]$ and $\theta_{ij} \in U[-4,4]$ (10 trials).

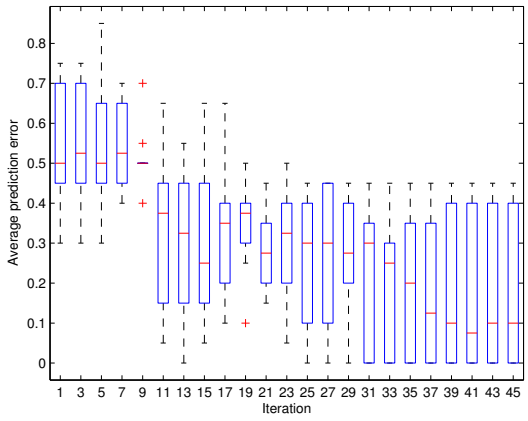

Figure 6: Convergence on 20 node Complete Graph with $\theta_i \in U[-1,1]$ and $\theta_{ij} \in U[-4,4]$ (10 trials).

cycle inequalities are satisfied. In each iteration we add to $\mathcal{R}$ at most[8] N violated cycle inequalities, coming from the N shortest paths found at each node of the graph. These experiments are using the TRW entropy approximation. In Figure 3 we show boxplots of the l_1 error for 10x10 grid MRFs over 40 trials, where $\theta_i \sim U[-1,1]$ and $\theta_{ij} \sim U[-4,4]$. The red line gives the median, and the blue boxes show the upper and lower quartiles. Iteration 1 corresponds to TRW with only the local consistency constraints. All of the cycle inequalities were satisfied within 10 iterations. After only 5 iterations (corresponding to solving the TRW objective 5 times, each time using a tighter relaxation of the marginal polytope) the median l_1 error in the singleton marginals dropped from over .35 to under .2.

[8]In practice, many of the cycles in G' are not simple cycles in G, so many fewer cycle inequalities are added.

In Figure 4 we look at whether the pseudomarginals are on the correct side of .5 – this gives us some idea of how much improvement our algorithm would give if were we to do classification based on the marginals found by approximate inference. We found the exact marginals using the Junction Tree algorithm. We observed the same convergence results on a 30x30 grid, although we could not access the accuracy due to the difficulty of exact marginals calculation. From these results, we expect that our algorithm will be both fast and accurate on larger structured models.

While these results are promising, real-world MRFs may have different structure, so we next looked at the other extreme. In Figures 5 and 6 we give analogous results for 20 node complete MRFs. In this difficult setting the algorithm took many more iterations before all cycle inequalities were satisfied, although the total number of cycle inequalities added was still significantly smaller than the number of triangle inequalities. While the improvement in the average l_1 error is roughly monotonic as the number of iterations increase, the change in the prediction accuracy is certainly not. Regardless, the eventual improvement in prediction accuracy is striking, with the median going from .5 (as bad as a coin flip) to .1.

5 CONCLUSION

We have demonstrated the value of cutting plane algorithms and cycle inequalities for obtaining tighter outer bounds on the marginal polytope. By better approximating the marginal polytope we were able to improve the accuracy of predicted marginal probabilities. The methods discussed in this paper have not yet been optimized for computational efficiency, hence we have not reported any running time comparisons. Our work raises several clear open questions that we hope to address in the follow-up work.

1. How to exploit graph structure in PSD conditions? E.g., iteratively adding linear inequalities that are violated by a solution which is not PSD.
2. How can we project the odd-wheel and bicycle odd-wheel inequalities to yield an efficient algorithm for sparse graphs?
3. Can we handle non-pairwise MRFs in ways other than mapping the MRF into a larger state space with pairwise interactions?
4. Can we bound the number of inequalities added for certain classes of MRFs?
5. Is it feasible to find the most violated constraint, i.e. the one which will decrease the objective function the most?

Acknowledgments

The authors thank Amir Globerson and David Karger for helpful discussions, and to A.G. for providing code for TRW.

References

[1] F. Barahona, *On cuts and matchings in planar graphs*, Mathematical Programming, 60 (1993), pp. 53–68.

[2] F. Barahona, M. Grotschel, M. Junger, and G. Reinelt, *An application of combinatorial optimization to statistical physics and circuit layout design*, Operations Research, 36 (1988), pp. 493–513.

[3] F. Barahona and A. Mahjoub, *On the cut polytope*, Mathematical Programming, 36 (1986), pp. 157–173.

[4] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.

[5] S. Chopra and J. Owen, *Extended formulations of the a-cut problem*, Mathematical Programming, 73 (1996), pp. 7–30.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd ed., 2001.

[7] M. M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, vol. 15 of Algorithms and Combinatorics, Springer, 1997.

[8] A. Globerson and T. Jaakkola, *Approximate inference using planar graph decomposition*, in Advances in Neural Information Processing Systems 20, 2007.

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

[10] ——, *A new class of upper bounds on the log partition function*, IEEE Transactions on Information Theory, 51 (2005), pp. 2313–2335.

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

[12] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families and variational inference*, Technical Report 649, UC Berkeley, Dept. of Statistics, 2003.

[13] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2001.

[14] C. Yanover, T. Meltzer, and Y. Weiss, *Linear programming relaxations and belief propagation – an empirical study*, JMLR Special Issue on Machine Learning and Large Scale Optimization, 7 (2006), pp. 1887–1907.

A GENERALIZATION

The cut polytope has a natural multi-cut formulation called the *A-partitions* problem. Suppose that every variable has at most m states. Given a MRF $G = (V, E)$ on n variables, construct the suspension graph $\nabla G = (V', E')$, where $V' = V \cup \{1, \ldots, m\}$[9], the additional m nodes corresponding to the m possible states. For each $v \in V$ having k possible states, we add edges (v, i) $\forall i = 1, \ldots, k$ to E' (which also contains all of the original edges E).

While earlier we considered cuts in the graph, now we must consider partitions $\pi = (V_1, V_2, \ldots, V_m)$ of the variables in V, where $v \in V_i$ signifies that variable v has state i. Let $E(\pi) \subset E'$ be the set of edges with endpoints in different sets of the partition (i.e. different assignments). Analogous to our definition of cut vectors (see Definition (1)) we denote $\delta(\pi)$ the vector of $\mathbb{R}^{E'}$ defined for $(i, j) \in E'$ by,

$$\delta(\pi)_{ij} = 1 \text{ if } (i, j) \in E(\pi), \text{ and 0 otherwise.} \quad (14)$$

The *multi-cut polytope* is the convex hull of the $\delta(\pi)$ vectors for all partitions π of the variables.

Chopra and Owen [5] define a relaxation of the multi-cut polytope analogous to the local consistency polytope. Although their formulation has exponentially many constraints (in m, the number of states), they show how to separate it in polynomial time, so we could easily integrate this into our cutting-plane algorithm. If G is a Potts model, then the minimal marginal polytope (i.e. having variables only for the minimal sufficient statistics) is in 1-1 correspondence with the multi-cut polytope.

This formulation gives an interesting trade-off when comparing the usual local consistency relaxation to the multi-cut analogue. In the former, the number

[9] As in the binary case, $n + m - 1$ nodes are possible, using a minimal representation. However, the mapping from the multi-cut polytope to the marginal polytope becomes more complex.

of variables are $O(m|V| + m^2|E|)$, while in the latter, the number of variables are $O(m|V| + |E|)$ but (potentially many) constraints need to be added by the cutting-plane algorithm. It would be interesting to see whether using the multi-cut relaxation significantly improves the running time of the LP relaxations of the Potts models in Yanover et al. [14], where the large number of states was a hindrance.

Chopra and Owen [5] also give a *per cycle* class of odd cycle inequalities (exponential in m and $|C|$, the cycle length), and show how to separate these in polynomial time (per cycle). It is not clear whether it is possible to separate all of the cycle inequalities in polynomial time for the multi-cut polytope. Regardless, we could always choose a small basis of cycles for which to run this separation oracle (e.g., the squares of a grid).

When given a MRF which is not a Potts model, the marginal polytope is in general not 1-1 with the multi-cut polytope; the linear mapping from the marginal polytope to the cut polytope is not injective. However, we can still optimize over the intersection of the local consistency polytope and the above relaxations of the multi-cut polytope. The linear mapping which is used by the algorithm is $x_{ij} = \sum_{a \neq b} \mu_{ij;ab}$.

B REMARKS ON COMPLEXITY

A natural question that is raised in this work is whether is is possible to efficiently test whether a point is in the marginal polytope.

Theorem 1. *The following decision problem is NP-complete: given a vector $\vec{\mu} \in \mathbb{R}_+^{V_n \cup E_n}$, decide if $\mu \in \mathcal{M}$.*

Proof. Using the linear bijection ξ, this problem is equivalent to the decision problem for $\mathrm{CUT}_n^{\square}$ (the same as ℓ_1-embeddability). The latter is shown to be NP-complete in [7]. □