Cutting Plane Algorithms for Variational Inference in Graphical Models

by

David Alexander Sontag

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007

Author .....
Department of Electrical Engineering and Computer Science
May 25, 2007

Certified by .....
Tommi S. Jaakkola
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Cutting Plane Algorithms for Variational Inference in Graphical Models

by

David Alexander Sontag

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

In this thesis, we give a new class of outer bounds on the marginal polytope, and propose a cutting-plane algorithm for efficiently optimizing over these constraints. When combined with a concave upper bound on the entropy, this gives a new variational inference algorithm for probabilistic inference in discrete Markov Random Fields (MRFs). Valid constraints are derived for the marginal polytope through a series of projections onto the cut polytope. Projecting onto a larger model gives an efficient separation algorithm for a large class of valid inequalities arising from each of the original projections. As a result, we obtain tighter upper bounds on the log-partition function than possible with previous variational inference algorithms. We also show empirically that our approximations of the marginals are significantly more accurate. This algorithm can also be applied to the problem of finding the Maximum a Posteriori assignment in a MRF, which corresponds to a linear program over the marginal polytope. One of the main contributions of the thesis is to bring together two seemingly different fields, polyhedral combinatorics and probabilistic inference, showing how certain results in either field can carry over to the other.

Thesis Supervisor: Tommi S. Jaakkola
Title: Associate Professor

Acknowledgments

My first two years at MIT have been a real pleasure, and I am happy to have so many great colleagues to work with. I particularly appreciate Leslie Kaelbling's guidance during my first year. Working with Bonnie Berger and her group on problems in computational biology has been very rewarding, and the problems we considered have helped provide perspective and serve as useful examples during this more theoretical work on approximate inference. I have also very much enjoyed being a part of the theory group.

Over the last year I have had the great opportunity to work with and be advised by Tommi Jaakkola, and this thesis is based on joint work with him. The dynamic of our conversations has been tons of fun, and I am looking forward to continuing work with Tommi over the next few years. I have also really enjoyed working with David Karger, and am grateful to David for initially suggesting that I look at the cutting-plane literature to tackle these inference problems. Amir Globerson has been my partner-in-crime for approximate inference research, and has been particularly helpful during this work, giving me the initial code for TRW and helping me debug various problems.

Thanks also to my office and lab mates for providing a stimulating and fun environment to work in, and to everyone who repeatedly inquired on my writing status and endeavored to get me to stop advancing the theory and start writing already. As always, the biggest thanks are due to my family for their constant support and patience. Finally, I thank Violeta for her love and support, for bearing with me during my busy weeks, and for making every day a happy one.

This thesis is dedicated in memory of Sean Jalal Hanna.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many interesting real-world problems can be approached, from a modeling perspective, by describing a joint probability distribution over a large number of variables. Over the last several years, graphical models have proven to be a valuable tool in both constructing and using these probability distributions. Undirected graphical models, also called *Markov Random Fields* (MRFs), are probabilistic models defined with respect to an undirected graph. The graph's vertices represent the variables, and separation in the graph is equivalent to conditional independence in the distribution. The probability distribution is specified by the product of non-negative *potential functions* on variables in the maximal cliques of the graph. The normalization term is called the *partition function*. Given some model, we are generally interested in two questions. The first is to find the marginal probabilities of specific subsets of the variables, and the second is to find the most likely setting of all the variables, called the *Maximum a Posteriori* (MAP) assignment. Both of these are intractable problems and require approximate methods.

Graphical models have been successfully applied to a wide variety of fields, from computer vision and natural language processing to computational biology. One of the many examples of their applications in computer vision is for image segmentation. Markov Random Fields for this problem typically have a variable for each pixel of the image, whose value dictates which segment it belongs to. Potentials are defined on adjacent pixels to enforce smoothness, discouraging pixels which look similar from

being assigned to different image segments. These models correspond to pairwise MRFs with non-binary variables. In computational biology, Sontag et al. [16] apply Bayesian networks to modeling systematic errors in high-throughput experiments for determining protein-protein interactions. This Bayesian network can be easily transformed into an equivalent non-pairwise MRF. The algorithms introduced in this thesis are directly applicable to inference problems in any discrete MRF, including the above-mentioned problems.

In this thesis we will focus on a particular class of approximate inference methods called *variational inference* algorithms. As we will show in Chapter 2, the *log-partition* function is convex in the model parameters, which allows us to derive a dual formulation consisting of a non-linear optimization over the *marginal polytope*, the set of marginal probabilities arising from valid MRFs with the same structure, i.e., marginal probabilities that are *realizable*. These marginal vectors act as the dual variables. For any marginal vector, the dual function is equal to the entropy of the maximum entropy distribution with those marginals. The marginal vector which maximizes this dual formulation gives the marginals of the MRF.

However, this formulation comes with its own difficulties. First, for graph structures other than trees, finding the entropy corresponding to any particular marginal vector is a hard problem. This has received much attention in recent years, and various approximations have been suggested. For example, in the tree-reweighted sum-product (TRW) algorithm of Wainwright et al. [17], the entropy is decomposed into a weighted combination of entropies of tree-structured distributions with the same pairwise marginals. When combined with an outer bound on the marginal polytope, this gives an upper bound on the log-partition function.

To add to the difficulty, the marginal polytope itself is hard to characterize. 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 A). 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. Most message-passing algorithms for evaluating marginals, including belief

propagation (sum product) and tree-reweighted sum-product, operate within the *local consistency polytope*, characterized by pairwise consistent marginals. For general graphs, the local consistency polytope is a *relaxation* of the marginal polytope.

We will show in Chapter 2 that finding the MAP assignment for MRFs can be cast as an integer linear program over the marginal polytope. Thus, any relaxations that we develop for the variational inference problem also apply to the MAP problem.

Cutting-plane algorithms are a well-known technique for solving integer linear programs, and are often used within combinatorial optimization. These algorithms typically begin with some relaxation of the solution space, and then find linear inequalities that separate the current fractional solution from all feasible integral solutions, iteratively adding these constraints into the linear program. The key to such approaches is to have an efficient separation algorithm which, given an infeasible solution, can quickly find a violated constraint, generally from a very large class of valid constraints on the set of integral solutions.

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. With each additional constraint, the relaxation becomes tighter. While earlier work focused on minimizing an upper bound on the log-partition function by improving entropy upper bounds, we minimize the upper bound on the log-partition function by improving the outer bound on the marginal polytope.

The motivation for our approach comes from the cutting-plane literature for the maximum cut problem. In fact, Barahona et al. [3] showed that the MAP problem in pairwise binary MRFs is equivalent to a linear optimization over the cut polytope, which is the convex hull of all valid graph cuts. The authors then went on to show how tighter relaxations on the cut polytope can be achieved by using a separation algorithm together with the cutting-plane methodology. While this work received significant exposure in the statistical physics and operations research communities, it went mostly unnoticed in the machine learning and statistics communities, possibly because few interesting MRFs involve only binary variables and pairwise potentials.

One of our main contributions is to derive a new class of outer bounds on the marginal polytope of non-binary and non-pairwise MRFs. The key realization is that valid constraints can be constructed by a series of *projections* onto the cut polytope. We then go on to show that projecting onto a *larger* graph than the original model leads to an efficient separation algorithm for these exponentially many projections. This result directly gives new variational inference and MAP algorithms for general MRFs, opening the door to a completely new direction of research for the machine learning community.

Another contribution of this thesis is to bring together two seemingly different fields, polyhedral combinatorics and probabilistic inference. By showing how to derive valid inequalities for the marginal polytope from any valid inequality on the cut polytope, and, similarly, how to obtain tighter relaxations of the multi-cut polytope using the marginal polytope as an extended formulation, we are creating the connection for past and future results in either field to carry over to the other. We give various examples in this thesis of results from polyhedral combinatorics that become particularly valuable for variational inference. Many new results in combinatorial optimization may also turn out to be helpful for variational inference. For example, in a recent paper [13], Krishnan et al. propose using cutting-planes for positive semi-definite constraints, and in future work are investigating how to do so while taking advantage of sparsity, questions which we also are very interested in. In Chapter 5 we go the other direction, introducing new valid inequalities for the marginal polytope, which, in turn, can be used in cutting-plane algorithms for multi-cut.

Chapter 2

Background

2.1 Exponential Family and Graphical Models

In this thesis, we consider inference problems in undirected graphical models, also called *Markov Random Fields* or *Markov networks*, that are probability distributions in the exponential family. For more details on this material, see the technical report by Wainwright and Jordan [18].

Let $\mathbf{x} \in \chi^n$ denote a random vector on n variables, where each variable x_i takes on the values in $\chi_i = \{0, 1, \ldots, m\}$. The exponential family is parameterized by a set of d *potentials* or *sufficient statistics* $\phi(\mathbf{x}) = \{\phi_i\}$ which are functions from χ^n to $\mathbb{R}$, a vector $\theta \in \mathbb{R}^d$, and the log-normalization (partition) function $A(\theta)$. The probability distribution can thus be written as:

$$p(\mathbf{x}; \theta) = \exp\{\langle \theta, \phi(\mathbf{x}) \rangle - A(\theta)\} \tag{2.1}$$

$$A(\theta) = \log \sum_{\mathbf{x} \in \chi^n} \exp\{\langle \theta, \phi(\mathbf{x}) \rangle\} \tag{2.2}$$

where $\langle \theta, \phi(\mathbf{x}) \rangle$ denotes the dot product of the parameters and the sufficient statistics.

The undirected graphical model $G = (V, E)$ has vertices V for the variables and edges E between all vertices whose variables are together in some potential. All of our results carry over to directed graphical models, or *Bayesian networks*, by the process of moralization, in which we add undirected edges between all parents of every variable

and make all edges undirected. Each conditional probability distribution becomes a potential function on its variables.

We assume that the model is fully parameterized, i.e. that the potentials are of the form $\phi_{S;x} = \delta(\mathbf{x}_S = x)$, where $S \subseteq V$ and x is an assignment of the variables in S. This delta function $\delta(\mathbf{x}_S = x) = 1$ if $\mathbf{x}_S = x$, and 0 otherwise. In *pairwise MRFs*, potentials are constrained to be on at most two variables (edges of the graph). We will make significant use of the following notation:

$$\mu_{i;s} = E_\theta[\delta_{i;s}] = p(x_i = s; \theta) \tag{2.3}$$

$$\mu_{ij;st} = E_\theta[\delta_{ij;st}] = p(x_i = s, x_j = t; \theta). \tag{2.4}$$

We will often refer to a *minimal representation* of the exponential family, where there is no linear combination $\langle a, \phi(\mathbf{x})\rangle$ equal to a constant. The advantage of working with minimal representations is that there is a unique parameter vector θ associated with every distribution in the family.

We will focus initially on Markov Random Fields (MRFs) with pairwise potentials and binary variables, and will then show how our results can be generalized to the non-pairwise and non-binary setting. The exponential family distribution with binary variables, i.e. $\chi_i = \{0, 1\}$, and pairwise potentials can be written in minimal form as:

$$\log p(\mathbf{x}; \theta) = \sum_{i \in V} \theta_i x_i + \sum_{(i,j) \in E} \theta_{ij} x_i x_j - A(\theta) \tag{2.5}$$

$$= \langle \theta, \phi(\mathbf{x})\rangle - A(\theta) \tag{2.6}$$

where the vector $\phi(\mathbf{x})$ of dimension $d = |V| + |E|$ collects together x_i for $i \in V$ and $x_i x_j$ for $(i, j) \in E$. This also known as the *Ising model* in statistical physics. We will denote $\mu_i = E[x_i]$ and $\mu_{ij} = E[x_i x_j]$.

The inference task is to evaluate the mean vector $\mu = E_\theta[\phi(\mathbf{x})]$. The log-partition function plays a critical part in the inference calculations. Two important properties

of the log-partition function are:

$$\frac{\partial A(\theta)}{\partial \theta_i} = E_\theta[\phi_i(\mathbf{x})] \tag{2.7}$$

$$\frac{\partial^2 A(\theta)}{\partial \theta_i \partial \theta_j} = E_\theta[\phi_i(\mathbf{x})\phi_j(\mathbf{x})] - E_\theta[\phi_i(\mathbf{x})]E_\theta[\phi_j(\mathbf{x})]. \tag{2.8}$$

Equation (2.8) shows that the Hessian of $A(\theta)$ is the covariance matrix of the the probability distribution. Since covariance matrices are positive semi-definite, this proves that $A(\theta)$ is a convex function in θ. Equation (2.7) shows that the gradient vector of $A(\theta)$ at a point θ' is the mean vector $\mu = E_{\theta'}[\phi(\mathbf{x})]$. These will form the basis of the variational formulation that we will develop in Section 2.3.

2.2 Exact and Approximate Inference

Suppose we have the following tree-structured distribution:

$$p(\mathbf{x};\theta) = \exp\left\{\sum_{i\in V}\sum_{s\in\chi_i}\theta_{i;s}\phi_{i;s}(\mathbf{x}) + \sum_{(i,j)\in E}\sum_{s_i\in\chi_i}\sum_{s_j\in\chi_j}\theta_{ij;s_is_j}\phi_{ij;s_is_j}(\mathbf{x}) - A(\theta)\right\} \tag{2.9}$$

To exactly solve for the marginals we need to compute the partition function, given in Equation (2.2), and then do the following summation:

$$\mu = \sum_{\mathbf{x}\in\chi^n} p(\mathbf{x};\theta)\phi(\mathbf{x}). \tag{2.10}$$

In general, there will be exponentially many terms in the summations, even with binary variables. One approach to solving this exactly, called *variable elimination*, is to try to find a good ordering of the variables such that the above summation decomposes as much as possible. Finding a good ordering for trees is easy: fix a root node and use a depth-first traversal of the graph. The *sum-product algorithm* is a dynamic programming algorithm for computing the partition function and marginals in tree-structured MRFs. The algorithm can be applied to general MRFs by first decomposing the graph into a *junction tree*, and then treating each maximal clique as

a variable whose values are the cross-product of the values of its constituent variables. Such schemes will have complexity exponential in the *treewidth* of the graph, which for most non-trees is quite large.

Sampling algorithms can be used to approximate the above expectations by considering only a small number of the terms in the summations [1]. One of the most popular sampling methods is Markov chain Monte Carlo (MCMC). A Markov chain is constructed whose stationary distribution is provably the probability distribution of interest. We can obtain good estimates of both the partition function and the marginals by running the chain sufficiently long to get independent samples. While these algorithms can have nice theoretical properties, in practice it is very difficult to prove bounds on the mixing time of these Markov chains. Even when they can be shown, often the required running time is prohibitively large.

Another approach is to try to obtain bounds on the marginals. In the ideal scenario, the algorithm would be able to continue improving the bounds for as long as we run it. Bound propagation [14] is one example of an algorithm which gives upper and lower bounds on the marginals. As we will elaborate below, variational methods allow us to get upper and lower bounds on the partition function, which together allow us to get bounds on the marginals.

2.3 Variational Methods

In this thesis we will focus on variational methods for approximating the log-partition function and marginals. The convexity of $A(\theta)$ suggests an alternative definition of the log-partition function, in terms of its Fenchel-Legendre conjugate [18]:

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \left\{ \langle \theta, \mu \rangle - B(\mu) \right\}, \tag{2.11}$$

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

polytope:

$$\mathcal{M} := \left\{\mu \in \mathbb{R}^d \mid \exists p(X) \text{ s.t. } \mu = E_p[\phi(\mathbf{x})]\right\} \tag{2.12}$$

The value $\mu^* \in \mathcal{M}$ that maximizes (2.11) is precisely the desired mean vector corresponding to θ. One way of deriving Equation (2.11) is as follows. Let Q be any distribution in the exponential family with sufficient statistics $\phi(x)$, let $\mu_Q = E_Q[\phi(x)]$ be the marginal vector for Q, and let $H(Q)$ be the entropy of Q. We have:

$$\begin{aligned}
D_{KL}(Q||P) &= \sum_{x\in X} Q(x) \log \frac{Q(x)}{P(x)} && (2.13)\\
&= -H(Q) - \sum_{x\in X} Q(x) \log P(x) && (2.14)\\
&= -H(Q) - \sum_{x\in X} Q(x)\langle\theta, \phi(x)\rangle + A(\theta) && (2.15)\\
&= -H(Q) - \langle\theta, \mu_Q\rangle + A(\theta) && (2.16)\\
&\geq 0. && (2.17)
\end{aligned}$$

Re-arranging, we get

$$\begin{aligned}
A(\theta) &\geq \langle\theta, \mu_Q\rangle + H(Q) && (2.18)\\
&= \langle\theta, \mu_Q\rangle + H(\mu_Q) && (2.19)
\end{aligned}$$

where $H(\mu_Q)$ is the maximum entropy of all the distributions with marginals μ_Q. We have an equality in the last line because the maximum entropy distribution with those marginals is Q, since Q is in the exponential family. Since this inequality holds for any distribution Q, and the marginal polytope $\mathcal{M}$ is the set of all valid marginals arising from some distribution, we have

$$A(\theta) \geq \sup_{\mu\in\mathcal{M}} \langle\theta, \mu\rangle + H(\mu). \tag{2.20}$$

Finally, since the inequality in (2.17) is tight if and only if $Q = P$, the inequality in (2.18) should be an equality, giving us (2.11). This also proves that the marginal

vector μ^* which maximizes (2.20) is equal to μ_P.

In general both $\mathcal{M}$ and the entropy $H(\mu)$ 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. The approximate mean vectors are called *pseudomarginals*. We will demonstrate later in the thesis that tighter outer bounds on $\mathcal{M}$ are valuable, especially for MRFs with large couplings θ_{ij}.

2.3.1 Naive Mean Field

Mean field algorithms try to find the distribution Q from a class of tractable distributions such that $D_{KL}(Q||P)$ is minimized. For example, in the naive mean field algorithm, we use the class of distributions where every node is independent of the others (fully disconnected MRFs). As can be seen from (2.20), this yields a lower bound on the log-partition function. This approximation corresponds to using an inner bound on the marginal polytope, since independence implies that the pairwise joint marginals are products of single variable marginals. The key advantage of using this inner bound is that, for these points in the marginal polytope, the entropy can be calculated exactly as the sum of the entropies of the individual variables. We thus get the following naive mean field objective:

$$A(\theta) \geq \sup_{\mu \in \mathcal{M}_{\text{naive}}} \langle \theta, \mu \rangle - \sum_{i \in V} \sum_{s \in \chi_i} \mu_{i;s} \log \mu_{i;s} \tag{2.21}$$

$$\mathcal{M}_{\text{naive}} = \left\{ \mu_{i;s} \in [0,1], \sum_{s \in \chi_i} \mu_{i;s} = 1, \mu_{ij;st} = \mu_{i;s}\mu_{j;t} \right\} \tag{2.22}$$

Although the objective is not convex, we can solve for a local optimum using gradient ascent or message-passing algorithms. The message-passing algorithms also have an interpretation in terms of a large sample approximation of Gibbs sampling for the model [18].

Since the approximating distribution is generally much simpler than the true distribution, and because we minimize $D_{KL}(Q||P)$ and not $D_{KL}(P||Q)$, mean field algorithms will attempt to exactly fit some of the modes of the true distribution, while

ignoring the rest. For example, if we were trying to find the Gaussian distribution which minimizes the KL-divergence to a mixture of Gaussians, we would converge to one of the mixture components. While this does yield a lower bound on the log-partition function, it may give a bad approximation of the marginals. An alternative, which we will explore extensively in this thesis, is to use an *outer bound* on the marginal polytope, allowing for more interesting approximating distributions, but paying the price of no longer having a closed form expression for the entropy.

2.3.2 Loopy Belief Propagation

One of the most popular variational inference algorithms is loopy belief propagation, which is the sum-product algorithm applied to MRFs with cycles.

Every marginal vector must satisfy local consistency, meaning that any two pairwise marginals on some variable must yield, on integration, the same singleton marginal of that variable. These constraints give us the local consistency polytope:

$$\text{LOCAL}(G) = \left\{ \mu \geq 0 \mid \sum_{s \in \chi_i} \mu_{i;s} = 1, \sum_{t \in \chi_j} \mu_{ij;st} = \mu_{i;s} \right\} \tag{2.23}$$

Since all marginals in $\mathcal{M}$ must satisfy (2.23), $\mathcal{M} \subseteq \text{LOCAL}(G)$, giving an outer bound on the marginal polytope. For tree-structured MRFs, these constraints fully characterize the marginal polytope, i.e. $\mathcal{M} = \text{LOCAL}(G)$. Furthermore, for general graphs, both LOCAL(G) and $\mathcal{M}$ have the same integral vertices [18, 11].

In general it is difficult to give $H(\mu)$ exactly, because there may be many different distributions that have the same marginal vector, each distribution having a different entropy. In addition, for $\mu \in \text{LOCAL(G)} \backslash \mathcal{M}$ it is not clear how to define $H(\mu)$. However, for trees, the entropy decomposes simply as the sum of the single node

entropies and the mutual information along each edge:

$$H(\mu_i) = -\sum_{s \in \chi_i} \mu_{i;s} \log \mu_{i;s} \tag{2.24}$$

$$I(\mu_{ij}) = \sum_{s \in \chi_i, t \in \chi_j} \mu_{ij;st} \log \frac{\mu_{ij;st}}{\mu_{i;s}\mu_{j;t}} \tag{2.25}$$

$$H_{Bethe}(\mu) = \sum_{i \in V} H(\mu_i) - \sum_{(i,j) \in E} I(\mu_{ij}) \tag{2.26}$$

For a graph with cycles, this is known as the *Bethe approximation* of the entropy, and is not concave for graphs other than trees.

Yedidia et al. [22] showed that the fixed points of loopy belief propagation correspond precisely to local stationary points of the following variational approximation of the log-partition function:

$$\langle \theta, \mu \rangle + H_{Bethe} \tag{2.27}$$

This formulation gives neither a lower or an upper bound on the log-partition function. However, at least intuitively, if the μ^* which optimizes (2.11) gives the true marginals, then when LOCAL(G) and H_{Bethe} are good approximations to $\mathcal{M}$ and $H(\mu)$, respectively, we may hope that the global optimum of (2.27) is close to μ^*. Indeed, belief propagation has been shown empirically to give very good approximations to the marginals for many MRFs with cycles.

2.3.3 Tree-Reweighted Sum-Product

One of the biggest problems with this formulation is that the Bethe entropy approximation is not concave, so finding the global optimum of (2.27) is difficult. Loopy belief propagation often does not converge in MRFs with tight loops and large coupling values (i.e. when θ_{ij} is large). While there are various alternatives to message-passing for doing the optimization, a different approach, given by Wainwright et al. [17], is to use a concave approximation to the entropy.

Suppose that $G = (V, E)$ is a pairwise MRF. Recall that $H(\mu)$ is the maximum

entropy of all the distributions with marginals μ. Ignoring the pairwise marginals for some of the edges can only increase the maximum entropy, since we are removing constraints. Recall also that the entropy of a tree-structured distribution is given by (2.26). Thus, if we were to consider $\mu(T)$ for some $T \subseteq E$ such that $G' = (V, T)$ is a tree, then $H(\mu(T))$ gives a concave upper bound on $H(\mu)$.

The convex combination of the upper bounds for each spanning tree of the graph is also an upper bound of $H(\mu)$. Minimizing this convex combination yields a tighter bound. Let $S(G)$ be the set of all spanning trees of the graph G. For any distribution τ over $S(G)$, let ρ_{ij} be the edge appearance probability for edge (i, j):

$$\rho_{ij} = \sum_{T \in S(G)} \tau(T) \mathbb{1}\left[(i, j) \in T\right]. \tag{2.28}$$

Given the edge appearance probabilities $\vec{\rho}$ corresponding to some τ, the sum of the exponentially many entropy terms (one for each spanning tree) can be collapsed into the following expression in terms of $\vec{\rho}$:

$$A(\theta) \leq \sup_{\mu \in \text{LOCAL}(G)} \langle \theta, \mu \rangle + \sum_{i \in V} H(\mu_i) - \sum_{(i,j) \in E} \rho_{ij} I(\mu_{ij}) \tag{2.29}$$

The set of $\vec{\rho}$ vectors that can arise from any distribution τ is the well-studied *spanning tree polytope*. For any fixed $\vec{\rho}$, the optimization in (2.29) can be done efficiently using the *tree-reweighted sum-product (TRW)* algorithm. The $\vec{\rho}$ vector can be optimized using conditional gradient together with a minimum spanning tree algorithm.

2.3.4 Log-determinant Relaxation

While the Bethe and TRW approximations to the entropy are based on tree decompositions, the log-determinant relaxation of Wainwright et al. [19] is based on a Gaussian approximation. We will state their result for Ising models; the generalization to multinomial states is given in [19]. First, we need a semi-definite outer bound on the marginal polytope. For any marginal vector μ on K_n, the complete graph with

n nodes, define:

$$M_1(\mu) \;=\; \begin{bmatrix} 1 & \mu_1 & \mu_2 & \cdots & \mu_{n-1} & \mu_n \\ \mu_1 & \mu_1 & \mu_{12} & \cdots & \mu_{1,n-1} & \mu_{1n} \\ \mu_2 & \mu_{21} & \mu_2 & \cdots & \mu_{2,n-1} & \mu_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{n-1} & \vdots & \vdots & \vdots & \vdots & \mu_{n-1,n} \\ \mu_n & \mu_{n1} & \mu_{n2} & \cdots & \mu_{n,n-1} & \mu_n \end{bmatrix}. \tag{2.30}$$

If μ is the marginal vector arising from some distribution $p(\mathbf{x};\theta)$, then $M_1(\mu) = E_\theta[(1\ \mathbf{x})^T(1\ \mathbf{x})]$ is the matrix of second moments for the vector $(1\ \mathbf{x})$ and is positive semi-definite. Thus, we obtain the following outer bound on the marginal polytope of complete graphs[1]:

$$\mathrm{SDEF}_1(K_n) \;=\; \left\{\mu \in \mathbb{R}^+ \mid M_1(\mu) \succeq 0\right\}. \tag{2.31}$$

The maximum (differential) entropy distribution of any continuous random vector with covariance $M_1(\mu)$ is the Gaussian distribution with the same covariance. Since we are interested in using this to obtain an upper bound on the *discrete entropy*, we define a continuous random vector $\tilde{\mathbf{x}} = \mathbf{x} + \vec{u}$, where $u_i \sim \mathcal{U}[-\frac{1}{2}, \frac{1}{2}]$. It is shown in [19] that $h(\tilde{\mathbf{x}}) = H(\mathbf{x})$, yielding the following upper bound to the log-partition function:

$$A(\theta) \leq \sup_{M_1(\mu)\succeq 0} \left\{ \langle \theta, \mu \rangle + \frac{1}{2} \log\det \left[M_1(\mu) + \frac{1}{12}\mathrm{blkdiag}[0, I_n] \right] \right\} + \frac{n}{2}\log(2\pi e) \tag{2.32}$$

We can improve the upper bound on $A(\theta)$ by using a tighter outer bound on the marginal polytope, e.g. $\mathrm{SDEF}_1(K_n) \cap \mathrm{LOCAL}(K_n)$. The $M_1(\mu) \succeq 0$ constraint is necessary for the entropy and log-partition upper bounds to hold. Wainwright et al. suggest relaxing this constraint, instead letting the $\log\det$ act as a barrier function to enforce the slightly weaker $M_1(\mu) \succeq -\frac{1}{12}\mathrm{blkdiag}[0, I_n]$ constraint; they are able to derive more efficient optimization algorithms in this setting.

[1] If a given MRF is incomplete, simply add variables for the remaining pairwise marginals.

Higher order moment matrices must also be positive semi-definite, leading to a sequence of tighter and tighter relaxations known as the Lasserre relaxations. However, this is of little practical interest since representing higher order moments would lead to an exponential number of variables in the relaxation. One of the conclusions from this thesis is that an entirely different set of valid constraints, to be introduced in Chapter 4, give more accurate pseudomarginals than the first order semi-definite constraints, while still taking advantage of the sparsity of the graph.

2.4 Maximum a Posteriori

The MAP problem is to find the assignment $\mathbf{x} \in \chi^n$ which maximizes $P(\mathbf{x}; \theta)$, or equivalently:

$$\begin{aligned} \max_{\mathbf{x} \in \chi^n} \log P(\mathbf{x}; \theta) &= \max_{\mathbf{x} \in \chi^n} \langle \theta, \phi(\mathbf{x}) \rangle - A(\theta) && (2.33) \\ &= \sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle - A(\theta) && (2.34) \end{aligned}$$

where the log-partition function $A(\theta)$ is a constant for the purpose of finding the maximizing assignment and can be ignored. The last equality comes from the fact that the optimal value of a linear program is attained at an extreme point or vertex, and the extreme points of the marginal polytope are simply the delta distributions on assignments $\mathbf{x} \in \chi^n$. When the MAP assignment $\mathbf{x}^*$ is unique, we have that the maximizing $\mu^* = \phi(\mathbf{x}^*)$.

In summary, both inferring marginals and the MAP assignments correspond to optimizing some objective over the marginal polytope $\mathcal{M}$.

Chapter 3

Cutting-Plane Algorithm

The main result in this thesis is the proposed algorithm given in Table 3. The algorithm alternates between solving for an upper bound of the log-partition function (see eqn. 2.11) and tightening the outer bound on the marginal polytope by adding valid constraints that are violated by the pseudomarginals at the optimum μ^*. We will discuss the actual constraints and separation algorithms in the following chapters; for now it suffices to know that the algorithm is able to efficiently separate an exponentially large class of valid constraints. In effect, we are using a significantly tighter relaxation to the marginal polytope than LOCAL(G) without having to explicitly represent all constraints. In Chapter 5 we show how to generalize this algorithm to non-pairwise and non-binary MRFs.

Our results are focused on the marginal polytope, not the entropy upper bound. Any approximation $B^*(\mu)$ of the entropy function can be used with our algorithm, as long as we can efficiently do the optimization given in line 3 of Table 3. In particular, we have investigated using the log-determinant and TRW entropy approximations. 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[1]. Second, the resulting objectives are convex, allowing for efficient optimization using conditional gradient or other methods.

[1]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.

1.	**(initialize)** $\mathcal{R} \leftarrow$ **LOCAL(G).**
2.	**Loop:**
3.	Solve optimization $\max_{\mu \in \mathcal{R}} \{\langle \theta, \mu \rangle - B^*(\mu)\}$.
4.	Construct ∇G and assign weights $w = \xi(\mu^*)$.
5.	Run separation algorithms from Table 4.2.1.
6.	Add violated inequalities to $\mathcal{R}$. If none, stop.

Table 3.1: Cutting-plane algorithm for probabilistic inference in binary pairwise MRFs. Let μ^* be the optimum of the optimization in line 3.

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 [19]. The disadvantage is that it would require explicitly representing all $O(n^2)$ μ_{ij} variables[2], which may be inefficient for large yet sparse MRFs.

When the algorithm terminates, we can use the last μ^* vector as an approximation to the single node and pairwise marginals. The results given in Chapter 6 use this method. An alternative would be to use the upper bounds on the partition function given by this algorithm, together with lower bounds obtained by a mean field algorithm, in order to obtain upper and lower bounds on the marginals [12].

The algorithm for MAP is the same, but excludes the entropy function in line 3. As a result, the optimization is simply a linear program. Since all integral vectors in the relaxation $\mathcal{R}$ are extreme points of the marginal polytope, if μ^* is integral when the algorithm terminates, then it is the MAP assignment.

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

Chapter 4

Cut Polytope

4.1 Polyhedral Results

In this section we will show that the marginal polytope for binary pairwise MRFs[1] is equivalent to the cut polytope, which has been studied extensively within the fields of combinatorial and polyhedral optimization [4, 2, 11]. This equivalence enables us to translate relaxations of the cut polytope into relaxations of the marginal polytope. Let $\mathcal{M}_{\{0,1\}}$ denote the marginal polytope for Ising models, which we will call the *binary marginal polytope*:

$$\mathcal{M}_{\{0,1\}} := \left\{ \mu \in \mathbb{R}^d \mid \exists p(X) \text{ s.t. } \begin{array}{l} \mu_i = E_p[X_i], \\ \mu_{ij} = E_p[X_i X_j] \end{array} \right\} \tag{4.1}$$

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,

$$\delta(S)_{ij} = 1 \text{ if } |S \cap \{i, j\}| = 1, \text{ and 0 otherwise.} \tag{4.2}$$

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*

[1] In the literature on cuts and metrics (e.g. [11]), the marginal polytope is called the *correlation polytope*, and is denoted by $\text{COR}_n^{\square}$.

projected onto G is the convex hull of the above cut vectors:

$$\mathrm{CUT}^{\square}(G) = \Big\{ \sum_{S \subseteq V_n} \lambda_S \delta(S) \mid \sum_{S \subseteq V_n} \lambda_S = 1 \text{ and } \lambda_S \geq 0 \text{ for all } S \subseteq V_n \Big\}. \tag{4.3}$$

The cut polytope for the complete graph on n nodes is denoted simply by $\mathrm{CUT}^{\square}_n$. 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 [11].

4.1.1 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 binary 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 $\mu \in \mathcal{M}_{\{0,1\}}$ to $x \in \mathrm{CUT}^{\square}(\nabla G)$ 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 (2.34) as a MAX-CUT problem[2]:

$$\sup_{\mu \in \mathcal{M}_{\{0,1\}}} \langle \theta, \mu \rangle \quad = \quad \max_{x \in \mathrm{CUT}^{\square}(\nabla G)} \langle \theta, \xi^{-1}(x) \rangle. \tag{4.4}$$

Furthermore, any valid inequality for the cut polytope can be transformed into a valid inequality for the binary 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 binary marginal polytope by using the mapping.

[2]The edge weights may be negative, so the Goemans-Williamson approximation algorithm does not directly apply.

4.1.2 Relaxations of the Cut Polytope

It is easy to verify that every cut vector $\delta(S)$ (given in equation 4.2) must satisfy the triangle inequalities: $\forall i, j, k$,

$$\delta(S)_{ik} + \delta(S)_{kj} - \delta(S)_{ij} \geq 0 \tag{4.5}$$

$$\delta(S)_{ij} + \delta(S)_{ik} + \delta(S)_{jk} \leq 2. \tag{4.6}$$

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[3]. The *semimetric polytope* $\mathrm{MET}_n^{\square}$ consists of those points $x \geq 0$ which satisfy the triangle inequalities. The projection of these $O(n^3)$ inequalities onto an incomplete graph is non-trivial and will be addressed in the next section. If, instead, we 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 can now apply the inverse mapping ξ^{-1} to obtain the corresponding relaxations for the binary marginal polytope:

$$\xi^{-1}(\mathrm{MET}_n^{\square}) = \left\{ \mu \in \mathbb{R}_+^d \mid \begin{array}{l} \forall\ i, j, k \in V, \\ \mu_{ik} + \mu_{kj} - \mu_k \leq \mu_{ij}, \text{ and} \\ \mu_i + \mu_j + \mu_k - \mu_{ij} - \mu_{ik} - \mu_{jk} \leq 1 \end{array} \right\} \tag{4.7}$$

$$\xi^{-1}(\mathrm{RMET}_n^{\square}) = \left\{ \mu \in \mathbb{R}_+^d \mid \begin{array}{l} \forall\ (i, j) \in E, \\ \mu_{ij} \leq \mu_i, \mu_{ij} \leq \mu_j \\ \mu_i + \mu_j - \mu_{ij} \leq 1 \end{array} \right\} \tag{4.8}$$

The $\xi^{-1}(\mathrm{RMET}_n^{\square})$ polytope is equivalent to LOCAL(G) (2.23) projected onto the variables $\mu_{i;1}$ and $\mu_{ij;11}$. Interestingly, the triangle inequalities suffice to describe $\mathcal{M}_{\{0,1\}}$, i.e. $\mathcal{M}_{\{0,1\}} = \xi^{-1}(\mathrm{MET}^{\square}(\nabla G))$, for a graph G if and only if G has no K_4-minor[4].

[3]Some authors call these *triplet constraints*.

[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 [11], CUT(G)=MET(G) when the graph has no K_5 minor, so it would be exact for planar Ising models with no field.

4.1.3 Cluster Relaxations and View from Cut Polytope

One approach to tightening the local consistency relaxation, used, for example, in generalized Belief Propagation (GBP) [23], is to introduce higher-order variables to represent the joint marginal of clusters of variables in the MRF. This improves the approximation in two ways: 1) it results in a tighter outer bound on the marginal polytope, and 2) these higher-order marginals can be used to get a better entropy approximation. In particular, if we had higher-order variables for every cluster of variables in the junction tree of a graph, it would exactly characterize the marginal polytope.

Exactly representing the joint marginal for a cluster of n variables is equivalent to the constraint that the projected marginal vector (onto just the variables of that cluster) belongs to the marginal polytope on n variables. Thus, for small enough n, an alternative to adding variables for the cluster's joint marginal would be to use the constraints corresponding to all of the facets of the corresponding binary marginal polytope. Deza and Laurent [11] give complete characterizations of the cut polytope for $n \leq 7$.

Triangle inequalities, corresponding to clusters on three variables, were proposed by various authors [19, 12] as a means of tightening the relaxation of the binary marginal polytope. However, they were added only for those edges already present in the MRF. The *cycle inequalities* that we will introduce in the next section include the triangle inequalities as a special case. The cutting plane algorithm given in Section 3, which separates all cycle inequalities, will result in at least as strong of a relaxation as the triangle inequalities would give.

This perspective allows us to directly compare the relaxation to the marginal polytope given by all triangle inequalities versus, for example, the square clusters used for grid MRFs. The cut polytope on 5 nodes (corresponding to the four variables in the square in addition to the suspension node) is characterized by 56 triangle inequalities and by *pentagonal inequalities* [11]. Thus, by just using cycle inequalities we are capturing the vast majority, but not all, of the facets induced by the cluster

variables. Furthermore, using the cluster variables alone misses out on all of the global constraints given by the remaining cycle inequalities.

4.2 Separation Algorithms

In this section we discuss various other well-known inequalities for the cut polytope, and show how these inequalities, though exponential in number, can be separated in polynomial time. These separation algorithms, together with the mapping from the cut polytope to the binary marginal polytope, form the basis of the cutting-plane algorithm given in the previous chapter.

Each algorithm separates a different class of inequalities. All of these inequalities arise from the study of the *facets* of the cut polytope. A facet is a polygon whose corners are vertices of the polytope, i.e. a maximal (under inclusion) face. The triangle inequalities, for example, are a special case of a more general class of inequalities called the hypermetric inequalities [11] for which efficient separation algorithms are not known. Another class, the Clique-Web inequalities, contains three special cases for which efficient separation are known: the cycle, odd-wheel, and bicycle odd-wheel inequalities.

4.2.1 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 [2], shows that the projected $\text{MET}_n^{\square}$ polytope can be defined in terms of cycle inequalities on

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

Table 4.1: Summary of separation algorithms for cut polytope.

just those edges in $G = (V, E)$:

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

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 $\mathrm{CUT}^{\square}(G)$ polytope [4].

In general there are exponentially many cycles and cycle inequalities for a graph G. However, Barahona and Mahjoub [4, 11] 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:

$$\sum_{e \in C \backslash F} x_e + \sum_{e \in F} (1 - x_e) \geq 1. \tag{4.9}$$

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, we have found a violated cycle inequality; otherwise, $\vec{x}$ satisfies all cycle inequalities. Using Dijkstra's shortest paths algorithm with a Fibonacci heap [9], the separation problem can be solved in time $O(n^2 \log n + n|E|)$.

4.2.2 Odd-wheel Inequalities

The odd-wheel (4.10) and bicycle odd-wheel (4.11) inequalities [11] 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:

$$x_{uv} + \sum_{e \in C} x_e - \sum_{i \in V_C} (x_{iu} + x_{iv}) \leq 0 \tag{4.10}$$

$$x_{uv} + \sum_{e \in C} x_e + \sum_{i \in V_C} (x_{iu} + x_{iv}) \leq 2|V_C| \tag{4.11}$$

where V_C refers to the vertices of cycle C. We give a sketch of the separation algorithm for the first inequality (see [11] 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 (4.10) 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.

4.2.3 Other Valid Inequalities

Another class of inequalities for the cut polytope are the negative-type inequalities [11], which are the same as the positive semi-definite constraints on the second moment matrix [19]. While these inequalities are not facet-defining for the cut polytope, they do provide a tighter outer bound than the local consistency polytope, and lead to an approximation algorithm for MAX-CUT with positive edge weights. If a matrix A is not positive semi-definite, a vector x can be found in $\mathrm{O}(n^3)$ 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 [5] provide a way of giving, in closed form, a hyperplane which separates the fractional solution from all integral solutions. These inequalities are applicable to MAP because any fractional solution must lie outside of the marginal polytope. We show in Appendix A 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.

Chapter 5

New Outer Bounds on the Marginal Polytope

In this chapter we give a new class of valid inequalities for the marginal polytope of non-binary and non-pairwise MRFs, and show how to efficiently separate this exponentially large set of inequalities. This contribution of our work has applicability well beyond machine learning and statistics, as these novel inequalities can be used within any of branch-and-cut scheme for the multi-cut problem. The key theoretical idea will be of projections from the marginal polytope onto the cut polytope[1].

The techniques of aggregation and projection as a means for obtaining valid inequalities are well-known in polyhedral combinatorics [11, 6, 7]. Given a linear projection $\Phi(\mathbf{x}) = A\mathbf{x}$, any valid inequality $\mathbf{c}'\Phi(\mathbf{x}) \leq 0$ for $\Phi(\mathbf{x})$ also gives the valid inequality $\mathbf{c}'A\mathbf{x} \leq 0$ for $\mathbf{x}$. Prior work used aggregation for the *nodes* of the graph. Our contribution is to show how aggregation of the *states* of each node of the graph can be used to obtain new inequalities for the marginal polytope.

We begin by motivating why new techniques are needed for this non-binary setting. Suppose we have the MRF in Figure 5-1 with variables taking values in $\chi = \{0, 1, 2\}$ and we have a projection from $\mu \in \mathcal{M}$ to cut variables given by $x_{ij} = \sum_{s,t \in \chi, s \neq t} \mu_{ij;st}$. Let $\mathbf{x}$ be the cut vector arising from the assignment $a = 0, b = 1, c = 2$. Does $\mathbf{x}$

[1]For convenience, the projections will actually be onto the binary marginal polytope $\mathcal{M}_{\{0,1\}}$. Since these are equivalent via the transformation in the previous chapter, we will use their names interchangeably.

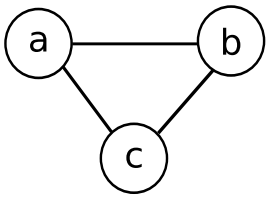


Figure 5-1: Triangle MRF

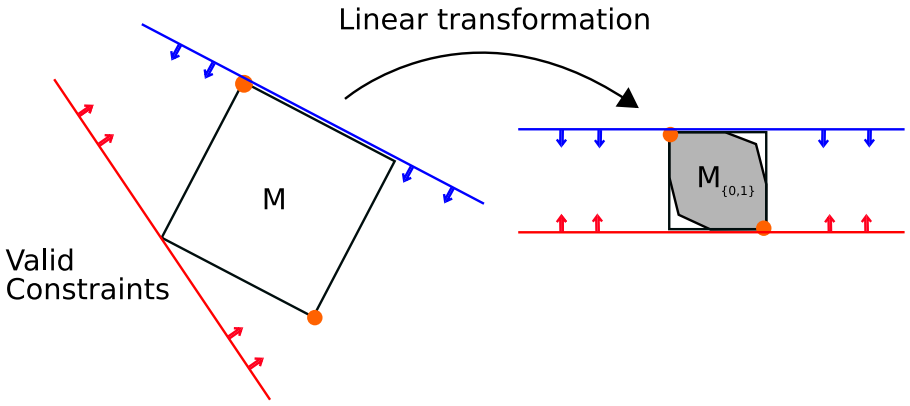


Figure 5-2: Illustration of projection from the marginal polytope of a non-binary MRF to the cut polytope of a different graph. All valid inequalities for the cut polytope yield valid inequalities for the marginal polytope, though not all will be facets. These projections map vertices to vertices, but the map will not always be onto.

satisfy the inequalities given in Chapter 4? While it does satisfy the first triangle inequality (4.5), it does not satisfy the second (4.6). In general, the cycle inequality (4.9) will hold only for $|F| = 1$. We call the convex hull of these cut vectors the *multi-cut polytope*. Although some inequalities have been given for the multi-cut polytope, discussed in Section 5.3, we find that, by considering the marginal polytope directly, we can construct a much richer class of inequalities.

Suppose $G = (V, E)$ is a pairwise MRF where each variable V_i takes on values in χ_i. For each variable, define the following *partition* of its values:

$$\pi_i : \chi_i \rightarrow \{0, 1\} \tag{5.1}$$

such that $\forall i, |\{s \in \chi_i \text{ s.t. } \pi_i(s) = 0\}| > 0$ and $|\{s \in \chi_i \text{ s.t. } \pi_i(s) = 1\}| > 0$. For any partition of all the variables π we define the following projection onto the cut polytope:

Definition 3. The linear map Ψ_π takes $\mu \in \mathcal{M}$ and for $i \in V$ assigns $\mu'_i = \sum_{s \in \chi_i \text{ s.t. } \pi_i(s)=1} \mu_{i;s}$ and for $(i,j) \in E$ assigns $\mu'_{ij} = \sum_{s_i \in \chi_i, s_j \in \chi_j \text{ s.t. } \pi_i(s_i)=\pi_j(s_j)=1} \mu_{ij;s_i s_j}$.

Each partition π gives a different projection, and there are $O(\prod_i 2^{|\chi_i|})$ possible partitions, or $O(2^{Nk})$ if all variables have k values. To construct valid inequalities for each projection we need to characterize the image space.

Theorem 1. *The image of the projection Ψ_π is $\mathcal{M}_{\{0,1\}}$, i.e. $\Psi_\pi : \mathcal{M} \to \mathcal{M}_{\{0,1\}}$. Furthermore, Ψ_π is surjective.*

Proof. Since Ψ_π is a linear map, it suffices to show that, for every extreme point $\mu \in \mathcal{M}$, $\Psi_\pi(\mu) \in \mathcal{M}_{\{0,1\}}$, and that for every extreme point $\mu' \in \mathcal{M}_{\{0,1\}}$, there exists some $\mu \in \mathcal{M}$ such that $\Psi_\pi(\mu) = \mu'$. The extreme points of $\mathcal{M}$ and $\mathcal{M}_{\{0,1\}}$ correspond one-to-one with assignments $\mathbf{x} \in \chi^n$ and $\{0,1\}^n$, respectively.

Given an extreme point $\mu \in \mathcal{M}$, let $\mathbf{x}'(\mu)_i = \sum_{s \in \chi_i \text{ s.t. } \pi_i(s)=1} \mu_{i;s}$. Since μ is an extreme point, $\mu_{i;s} = 1$ for exactly one value s, which implies that $\mathbf{x}'(\mu) \in \{0,1\}^n$. Then, $\Psi_\pi(\mu) = E[\phi(\mathbf{x}'(\mu))]$, showing that $\Psi_\pi(\mu) \in \mathcal{M}_{\{0,1\}}$.

Given an extreme point $\mu' \in \mathcal{M}_{\{0,1\}}$, let $\mathbf{x}'(\mu')$ be its corresponding assignment. For each variable i, choose some $s \in \chi_i$ such that $\mathbf{x}'(\mu')_i = \pi_i(s)$, and assign $\mathbf{x}_i(\mu') = s$. The existence of such s is guaranteed by our construction of π. Defining $\mu = E[\phi(\mathbf{x}(\mu'))] \in \mathcal{M}$, we have that $\Psi_\pi(\mu) = \mu'$. □

We will now give a more general class of projections, where we map the marginal polytope to a cut polytope of a *larger* graph. The projection scheme is general, and we will propose various classes of graphs which might be good candidates to use with it. A cutting-plane algorithm may begin by projecting onto a smaller graph, then advancing to projecting onto larger graphs only after satisfying all inequalities given by the smaller one.

Let $\pi_{\mathbf{i}} = \{\pi_i^1, \pi_i^2, \ldots\}$ be some *set of partitions* of node i. Every node can have a different number of partitions. Define the *projection graph* $G_\pi = (V_\pi, E_\pi)$ where there

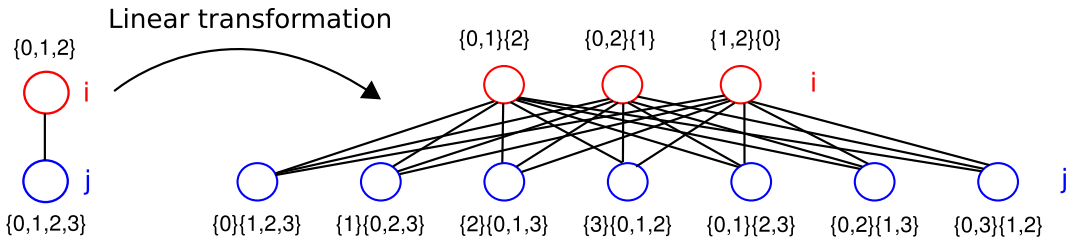


Figure 5-3: Illustration of the general projection Ψ_{G_π} for one edge $(i,j) \in E$ where $\chi_i = \{0,1,2\}$ and $\chi_j = \{0,1,2,3\}$. The projection graph G_π is shown on the right, having three partitions for i and seven for j.

is a node for every partition:

$$V_\pi = \bigcup_{i \in V} \pi_{\mathbf{i}} \tag{5.2}$$

$$E_\pi \subseteq \{(\pi_i^q, \pi_j^r) \mid (i,j) \in E, q \leq |\pi_{\mathbf{i}}|, r \leq |\pi_{\mathbf{j}}|\}. \tag{5.3}$$

Definition 4. The linear map Ψ_{G_π} takes $\mu \in \mathcal{M}$ and for each node $v = \pi_i^q \in V_\pi$ assigns $\mu'_v = \sum_{s \in \chi_i \text{ s.t. } \pi_i^q(s)=1} \mu_{i;s}$ and for each edge $e = (\pi_i^q, \pi_j^r) \in E_\pi$ assigns $\mu'_e = \sum_{s_i \in \chi_i, s_j \in \chi_j \text{ s.t. } \pi_i^q(s_i)=\pi_j^r(s_j)=1} \mu_{ij;s_i s_j}$.

This projection is a generalization of the earlier projection, where we had $|\pi_{\mathbf{i}}| = 1$ for all i. We call the former the *single projection graph*. We call the graph consisting of all possible node partitions and all possible edges the *full projection graph* (see Figure 5-3).

Let $\mathcal{M}_{\{0,1\}}(G_\pi)$ denote the binary marginal polytope of the projection graph.

Theorem 2. *The image of the projection Ψ_{G_π} is $\mathcal{M}_{\{0,1\}}(G_\pi)$, i.e. $\Psi_\pi : \mathcal{M} \to \mathcal{M}_{\{0,1\}}(G_\pi)$.*

Proof. Since Ψ_{G_π} is a linear map, it suffices to show that, for every extreme point $\mu \in \mathcal{M}$, $\Psi_{G_\pi}(\mu) \in \mathcal{M}_{\{0,1\}}(G_\pi)$. The extreme points of $\mathcal{M}$ correspond one-to-one with assignments $\mathbf{x} \in \chi^n$. Given an extreme point $\mu \in \mathcal{M}$ and variable $v = \pi_i^q \in V_\pi$, define $\mathbf{x}'(\mu)_v = \sum_{s \in \chi_i \text{ s.t. } \pi_i^q(s)=1} \mu_{i;s}$. Since μ is an extreme point, $\mu_{i;s} = 1$ for exactly one value s, which implies that $\mathbf{x}'(\mu) \in \{0,1\}^{|V_\pi|}$. Then, $\Psi_{G_\pi}(\mu) = E[\phi(\mathbf{x}'(\mu))]$, showing that $\Psi_{G_\pi}(\mu) \in \mathcal{M}_{\{0,1\}}(G_\pi)$. □

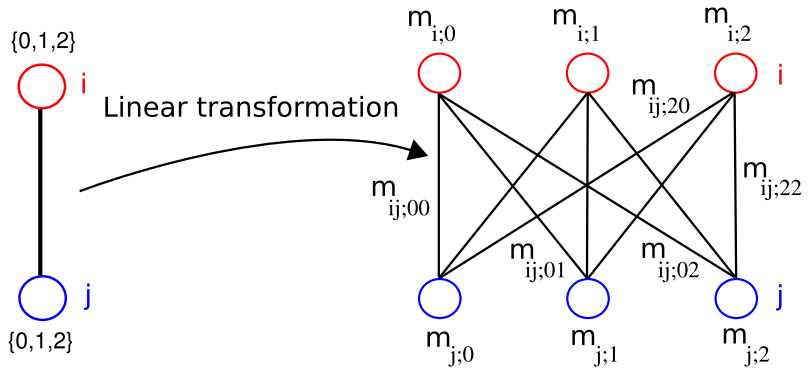


Figure 5-4: Illustration of the $k-$*projection graph* for one edge $(i,j) \in E$, where $\chi_i = \{0,1,2\}$. The nodes and (some of) the edges are labeled with the values given to them by the linear mapping, e.g. $\mu_{i;0}$ or $\mu_{ij;02}$.

In general the projection Ψ_{G_π} will not be surjective. Suppose every variable has k states. The simple projection graph has one 1 node per variable (and is surjective). The full projection graph has $O(2^k)$ nodes per variable. We illustrate in Figures 5-4 and 5-5 two other projection graphs, the first having k nodes per variable, and the second having $\log k$ nodes per variable. More specifically, define the $k-$*projection graph* $G_k = (V_k, E_k)$ where there is a node for each state of each variable:

$$V_k = \{v_{i;s} \mid i \in V, s \in \chi_i\} \tag{5.4}$$

$$E_k = \{(v_{i;s}, v_{j;t}) \mid (i,j) \in E, s \in \chi_i, t \in \chi_j\} \tag{5.5}$$

Definition 5. The linear map Ψ_k takes $\mu \in \mathcal{M}$ and for each node $v_{i;s} \in V_k$ assigns $\mu'_v = \mu_{i;s}$ and for each edge $(v_{i;s}, v_{j;t})$ assigns $\mu'_e = \mu_{ij;st}$.

We could also have defined this projection by giving the corresponding partitions and using Definition 4. Thus, the result from Theorem 2 applies.

The $\log k-$*projection graph* $G_{\log k}$ has $\log|\chi_i|$ partitions for each variable[2]. Let $b(s)_q$ be the q'th bit in the binary representation of $s \in \mathbb{Z}_+$. The partitions are defined as

$$\pi_i^q = \{s \in \chi_i \mid b(s)_q = 1\} \tag{5.6}$$

[2]Assume without loss of generality that $|\chi_i|$ is a power of 2.

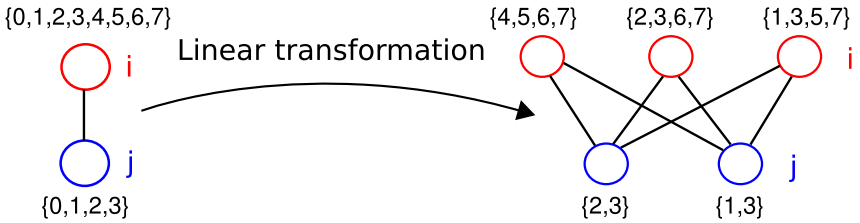


Figure 5-5: Illustration of the $\log k-$*projection graph* for one edge $(i,j) \in E$, where $\chi_i = \{0,1,2,3,4,5,6,7\}$ and $\chi_j = \{0,1,2,3\}$. Only half of each node's partition is displayed; the remaining states are the other half. The q'th partition arises from the q'th bit in the states' binary representation.

and the projection is given by Definition 4. The $\log k-$projection graph is interesting because the extreme points of $\mathcal{M}$ are one-to-one with the extreme points of its image. However, the linear map is not a bijection.

Theorem 3. *Assume $|\chi_i|$ is a power of 2 for all variables i. Then, the projection $\Psi_{G_{\log k}}$ is surjective. Furthermore, the extreme points of $\mathcal{M}$ are one-to-one with the extreme points of $\mathcal{M}_{\{0,1\}}$.*

Proof. We already showed in Theorem 2 that extreme points of $\mathcal{M}$ map to extreme points of $\mathcal{M}_{\{0,1\}}$. Given an extreme point $\mu' \in \mathcal{M}_{\{0,1\}}$, let $\mathbf{x}'(\mu')$ be its corresponding assignment. For each variable i, let $\mathbf{x}'(\mu')_i$ be the assignment to the $\log |\chi_i|$ nodes of variable i. Now consider $\mathbf{x}'(\mu')_i$ to be the binary expansion of the integer s, and assign $\mathbf{x}_i(\mu') = s$. Defining $\mu = E[\phi(\mathbf{x}(\mu'))] \in \mathcal{M}$, we have that $\Psi_{G_{\log k}}(\mu) = \mu'$. □

5.1 Separation Algorithm

We are now in the position to combine these projections with the cutting-plane algorithm from Chapter 3. The new algorithm is given in Table 5.1. Once we project the solution to the binary marginal polytope, any of the separation algorithms from Chapter 4 can be applied. This yields a new class of cycle inequalities and odd-wheel inequalities for the marginal polytope.

Consider the single projection graph G_π given by the (single) projection π. Suppose[3] that we have a cycle C in G and any $F \subseteq C$, $|F|$ odd. We obtain the following

[3]We could also derive cycle inequalities for the suspension graph ∇G_π. However, we omit this

1.	**(initialize)** $\mathcal{R} \leftarrow$ **LOCAL(G).**
2.	**Loop:**
3.	Solve optimization $\max_{\mu \in \mathcal{R}} \{\langle \theta, \mu \rangle - B^*(\mu)\}$.
4.	Choose a projection graph G_π, and let $\mu' = \Psi_{G_\pi}(\mu^*)$.
5.	Construct ∇G_π and assign weights $w = \xi(\mu')$.
6.	Run separation algorithms from Table 4.2.1.
7.	Add violated inequalities to $\mathcal{R}$. If none, stop.

Table 5.1: Cutting-plane algorithm for probabilistic inference in non-binary MRFs.

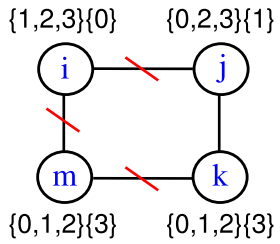


Figure 5-6: Illustration of the *single projection graph* G_π for a square graph, where all variables have states $\{0, 1, 2, 3\}$. The three red lines indicate an invalid cut; every cycle must be cut an even number of times.

valid inequality for $\mu \in \mathcal{M}$ by applying the projection Ψ_π and a cycle inequality:

$$\sum_{(i,j) \in C \setminus F} \mu_{ij}^\pi(x_i \neq x_j) + \sum_{(i,j) \in F} \mu_{ij}^\pi(x_i = x_j) \geq 1, \tag{5.7}$$

where we define:

$$\mu_{ij}^\pi(x_i \neq x_j) = \sum_{s_i \in \chi_i, s_j \in \chi_j \text{ s.t. } \pi_i(s_i) \neq \pi_j(s_j)} \mu_{ij;s_i s_j} \tag{5.8}$$

$$\mu_{ij}^\pi(x_i = x_j) = \sum_{s_i \in \chi_i, s_j \in \chi_j \text{ s.t. } \pi_i(s_i) = \pi_j(s_j)} \mu_{ij;s_i s_j}. \tag{5.9}$$

Consider the projection graph shown in Figure 5-6 and the corresponding cycle inequality, where F is illustrated by cut edges (in red). The following is an example

generalization for reasons of clarity.

of an extreme point of LOCAL(G) which is violated by this cycle inequality:

$$\begin{aligned} \mu_{i;0} = \mu_{i;3} = .5, \quad \mu_{j;1} = \mu_{j;2} = .5, \quad \mu_{m;1} = \mu_{m;3} = .5, \quad \mu_{k;2} = \mu_{k;3} = .5 \\ \mu_{ij;02} = \mu_{ij;31} = .5, \qquad \mu_{im;01} = \mu_{im;33} = .5 \\ \mu_{jk;13} = \mu_{jk;22} = .5, \qquad \mu_{mk;13} = \mu_{mk;32} = .5 \end{aligned} \tag{5.10}$$

This example shows that single projection graphs yield non-trivial inequalities.

Theorem 4. *For every single projection graph G_π and every cycle inequality arising from a chordless circuit C on G_π such that $|C| > 3$, $\exists \mu \in \mathbf{LOCAL}(G) \backslash \mathcal{M}$ such that μ violates that inequality.*

Proof. For each variable $i \in V$, choose s_i, t_i s.t. $\pi_i(s_i) = 1$ and $\pi_i(t_i) = 0$. Assign $\mu_{i;q} = 0$ for $q \in \chi_i \backslash \{s_i, t_i\}$. Similarly, for every $(i,j) \in E$, assign $\mu_{ij;qr} = 0$ for $q \in \chi_i \backslash \{s_i, t_i\}$ and $r \in \chi_j \backslash \{s_j, t_j\}$. The polytope resulting from the projection of $\mathcal{M}$ onto the remaining variables is equivalent to $\mathcal{M}_{\{0,1\}}$ for the same graph. Barahona and Mahjoub [4] showed that the cycle inequality on this chordless circuit is facet-defining for the cut polytope on ∇G_π, which is equivalent to $\mathcal{M}_{\{0,1\}}$ by a linear bijection. The projection of the local consistency constraints give the rooted triangle inequalities for ∇G_π, which, since $|C| > 3$, correspond to different facets of the cut polytope. If there does not exist such a μ then it implies that $\mathrm{CUT}^\square(\nabla G_\pi) = \mathrm{RMET}^\square(\nabla G_\pi)$, which is a contradiction of this inequality being facet-defining. □

This does not, however, say anything about the tightness of the relaxation resulting from all cycle inequalities, other than that it is strictly tighter than LOCAL(G). If all N variables have k values, then there are $O((2^k)^N)$ different single projection graphs. Instead of attempting to separate each graph individually, it suffices to consider just the full projection graph. Thus, even though the projection Ψ_{G_k} is not surjective, the full projection graph allows us to efficiently obtain a tighter relaxation than any of the other projection graphs in combination would give.

Theorem 5. *Suppose the number of values per node, k, is a constant. The separation problem of all cycle inequalities* (5.7) *for all single projection graphs, when we allow*

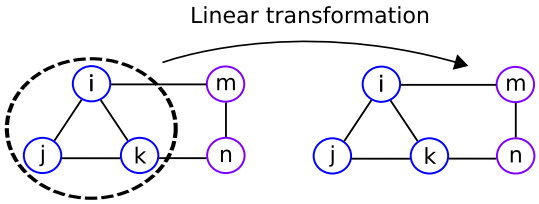


Figure 5-7: Example of a projection of a marginal vector from a non-pairwise MRF to the pairwise MRF on the same variables. The original model, shown on the left, has a potential on the variables i, j, k.

some additional valid inequalities for $\mathcal{M}$, can be solved in polynomial time.

Proof. All cycles in all single projection graphs are also found in the full projection graph. Thus, by separating all cycle inequalities for the full projection graph, which has $N2^k$ nodes, we get a strictly tighter relaxation. We showed in Chapter 4 that the separation problem of cycle inequalities for the binary marginal polytope can be solved in polynomial time in the size of the graph. □

5.2 Non-pairwise Markov Random Fields

The results from the previous section can be trivially applied to non-pairwise MRFs by first projecting onto a pairwise MRF, then applying the algorithm in Table 5.1. For example, the MRF in Figure 5-7 has a potential on the variables i, j, k, so the marginal polytope will have the variables $\mu_{ijk;stw}$ for $s \in \chi_i, t \in \chi_j, w \in \chi_k$. After projection, we will have the pairwise variables $\mu_{ij;st}$, $\mu_{jk;tw}$, and $\mu_{ik;sw}$. We can expect that the pairwise projection will be particularly valuable for non-pairwise MRFs where the overlap between adjacent potentials is only a single variable.

We can generalize the results of the previous section even further by considering clusters of nodes. Suppose we include additional variables, corresponding to the joint probability of a cluster of variables, to the marginal polytope. Figure 5-7 is an example where i, j, k were necessarily clustered because of appearing together in a potential. The cluster variable for i, j, k is a discrete variable taking on the values $\chi_i \times \chi_j \times \chi_k$.

We need to add constraints enforcing that all variables in common between two

clusters C_o and C_p have the same marginals. Let $V_c = C_o \cap C_p$ be the variables in common between the clusters, and let $V_o = C_o \backslash V_c$ and $V_p = C_p \backslash V_c$ be the other variables. Define χ_c to be all possible assignments to the variables in the set V_c. Then, for $\mathbf{x} \in \chi_c$, include the constraint:

$$\sum_{\mathbf{y} \in \chi_o} \mu_{C_o;\mathbf{x}\cdot\mathbf{y}} = \sum_{\mathbf{z} \in \chi_p} \mu_{C_p;\mathbf{x}\cdot\mathbf{z}}. \tag{5.11}$$

For pairwise clusters this is simply the usual local consistency constraints. We can now apply the projections of the previous section, considering various partitions of each cluster variable, to obtain a tighter relaxation of the marginal polytope.

5.3 Remarks on Multi-Cut Polytope

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 pairwise MRF $G = (V, E)$ on n variables, construct the suspension graph $\nabla G = (V', E')$, where $V' = V \cup \{1, \ldots, m\}$[4], 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,

$$\delta(\pi)_{ij} = 1 \text{ if } (i, j) \in E(\pi), \text{ and 0 otherwise.} \tag{5.12}$$

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

[4] 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.

variables.[5]

Chopra and Owen [8] define a relaxation of the multi-cut polytope analogous to the local consistency polytope. Although their formulation has exponentially many constraints (in m, the maximum 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 non-binary pairwise MRF which only has potentials of the form $\phi_{ij} = \delta(x_i \neq x_j)$, called a *Potts model*, then the marginal polytope is in one-to-one 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 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. [21], where the large number of states was a hindrance.

When given a MRF which is not a Potts model, the marginal polytope is in general not one-to-one with the multi-cut polytope; the linear mapping from the marginal polytope to the multi-cut polytope is not injective. The results of the previous sections can be generalized by projecting to the multi-cut polytope instead of the cut polytope. The linear mapping $x_{ij} = \sum_{a \neq b} \mu_{ij;ab}$ would carry over valid inequalities for the multi-cut polytope to the marginal polytope.

Chopra and Owen [8] give a *per cycle* class of odd cycle inequalities (exponential in m) for the multi-cut polytope, and show how to separate these in polynomial time (per cycle). These cycle constraints are different from the cycle constraints that we derived in the previous section – among other differences, these constraints are for cycles of length at most m. The authors were not able to come up with an algorithm to separate all of their cycle inequalities in polynomial time. One open question is whether these cycle inequalities can be derived from our projection scheme, which

[5] This definition is consistent with, and slightly more general than, the definition that we gave in the beginning of the chapter.

would yield an efficient separation algorithm.

Various other valid inequalities have been found for the multi-cut polytope. Deza et al. [10] generalize the clique-web inequalities to the multi-cut setting and show how, for the special case of odd-wheel inequalities, they can be separated in polynomial time. Borndörfer et al. [6] derive new inequalities for the multi-cut problem by reductions to the stable set problem. In particular, their reductions give a polynomial time algorithm for separating 2-chorded cycle inequalities.

If our original goal were to solve the multi-cut problem, the marginal polytope $\mathcal{M}$ could be considered an *extended formulation* of the original LP relaxations in which we add more variables in order to obtain tighter relaxations. However, while this directly gives an algorithm for solving multi-cut problems, actually characterizing the implicit constraints on the multi-cut polytope is more difficult.

Chapter 6

Experiments

We experimented with the algorithm shown in Table 3 for both MAP and marginals. We used the glpkmex and YALMIP [15] optimization packages within Matlab, and wrote the separation algorithms in Java. We made no attempt to optimize our code and thus omit running times. All of the experiments are on binary pairwise MRFs; we expect similar results for non-binary and non-pairwise MRFs.

6.1 Computing Marginals

In this section we show that using our algorithm to optimize over the $\xi^{-1}(\text{MET}_n^{\square})$ polytope yields significantly more accurate pseudomarginals than can be obtained by optimizing over LOCAL(G). We experiment with both the log-determinant [19] and the TRW [17] 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 [20]. We optimize the TRW objective with conditional gradient, using linear programming at each iteration to do the projection onto $\mathcal{R}$.

These trials were on pairwise MRFs with $x_i \in \{-1, 1\}$ (see eqn. 2.5) and mixed potentials. In Figure 6-1 we show results for 10 node complete graphs with $\theta_i \sim$

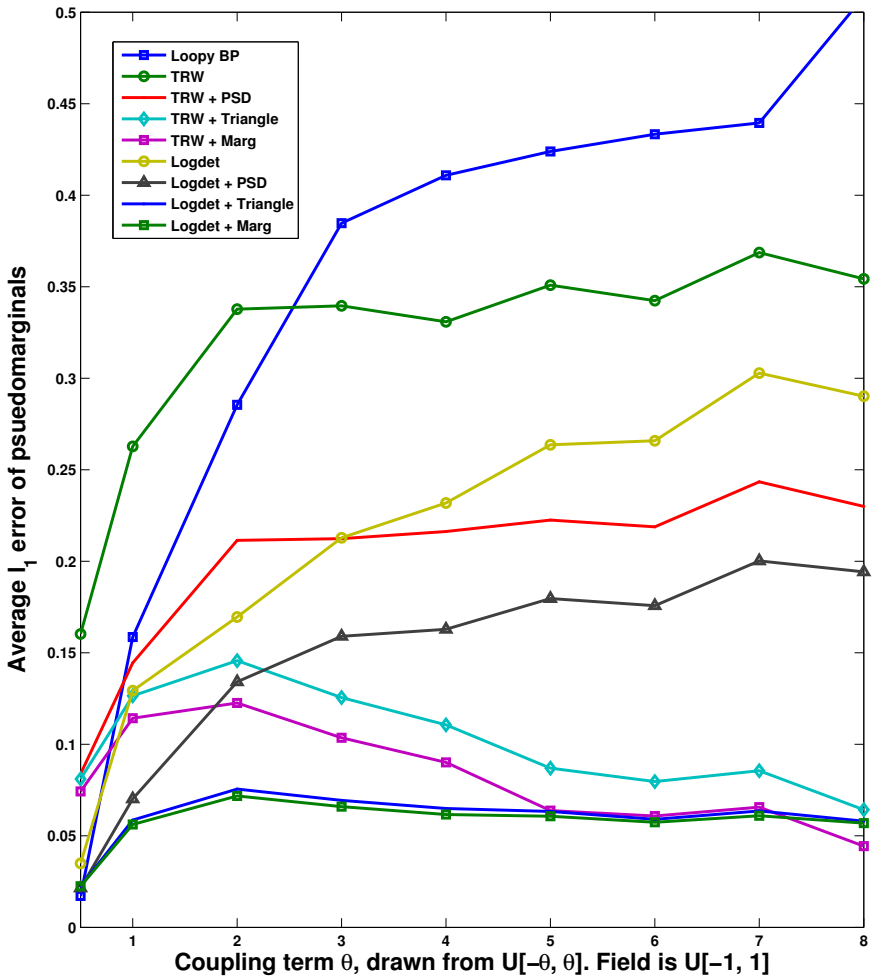


Figure 6-1: Accuracy of pseudomarginals on 10 node complete graph (100 trials).

$U[-1,1]$ 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 100 trials. The y-axis shows the average ℓ_1 error of the single node marginals. Note that although the coupling is so large, the external field is also significant, and the actual probabilities are interesting, bounded away from .5 and not all the same (as you would find in a highly coupled model with attractive potentials).

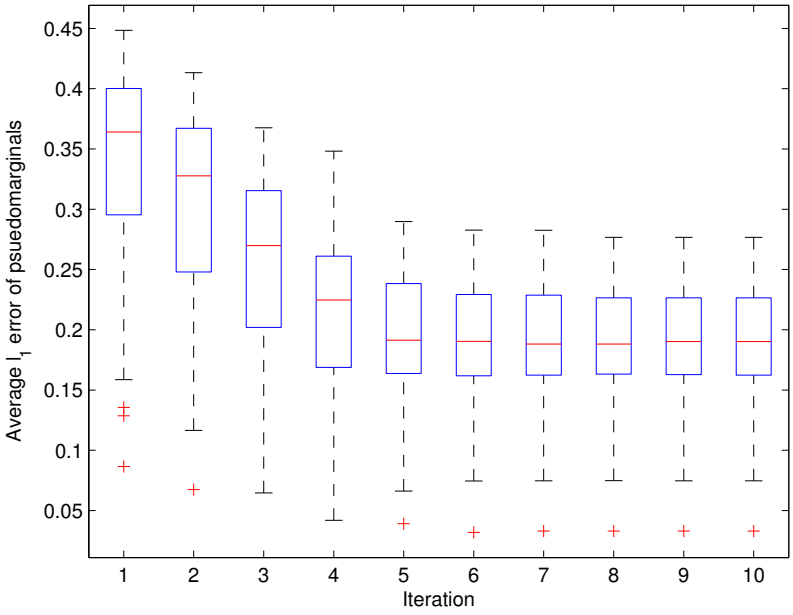
In this difficult setting, loopy belief propagation (with a .5 decay rate) seldom

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).

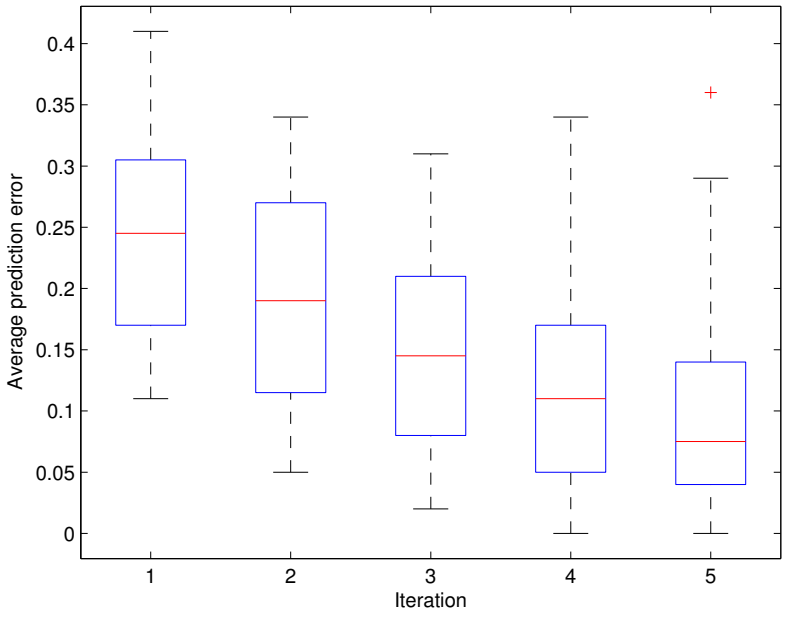
We were also interested in investigating the extent to which further tightening of the marginal polytope relaxations would improve pseudomarginal accuracy. The marginal polytope has 2^N vertices, where N is the number of variables in the binary MRF. Thus, for these small MRFs we can exactly represent the marginal polytope as the convex hull of its vertices. We show in Figure 6-1 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. Thus, for these entropy approximations, our algorithm may give as good results as can be hoped for. However, these results are dependent on what entropy approximation is used. For example, for a few 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 3) the algorithm takes before all cycle inequalities are satisfied. In each iteration we add to $\mathcal{R}$ at most[1] 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 6-2(a) 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,

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

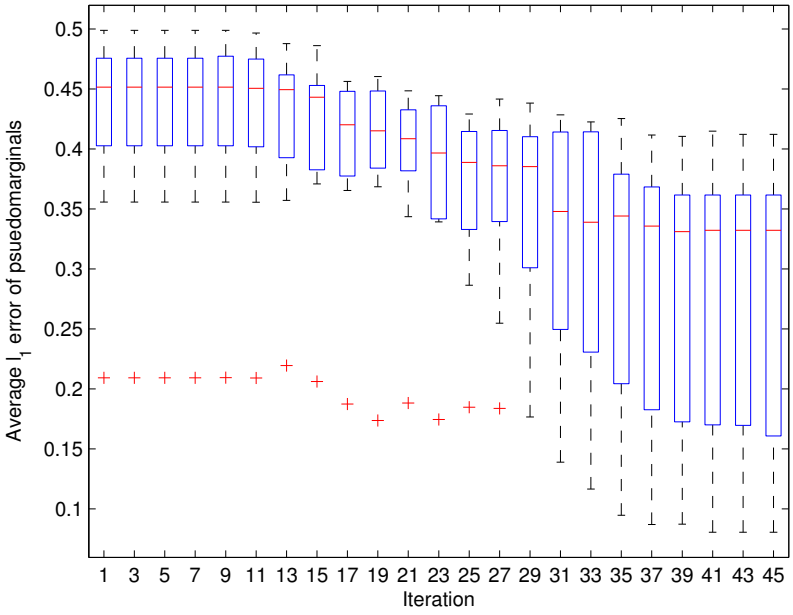


(a) Average ℓ_1 error

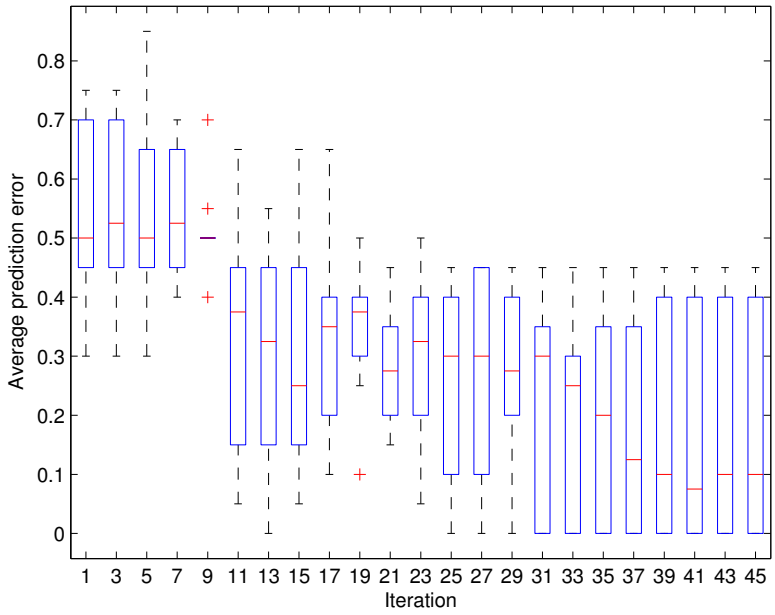


(b) Average prediction error

Figure 6-2: Convergence of cutting-plane algorithm with TRW entropy on 10x10 grid with $\theta_i \in U[-1, 1]$ and $\theta_{ij} \in U[-4, 4]$ (40 trials).



(a) Average ℓ_1 error



(b) Average prediction error

Figure 6-3: Convergence of cutting-plane algorithm with TRW entropy on 20 node complete graph with $\theta_i \in U[-1, 1]$ and $\theta_{ij} \in U[-4, 4]$ (10 trials).

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 single node marginals dropped from over .35 to under .2. In Figure 6-2(b) 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 we were to do classification using the marginals found by approximate inference. We calculated the exact marginals using the Junction Tree algorithm. We observed the same convergence results on a 30x30 grid, although we could not assess the accuracy due to the difficulty of exact marginals calculation. From these results, we predict 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 6-3(a) and 6-3(b) 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. The total number of cycle inequalities added was still significantly smaller than the number of triangle inequalities on the complete graph. 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.

6.2 Maximum a Posteriori

Applying our algorithm for MAP to Ising models gives the setting already considered by Barahona et al. [3]. We give experimental results here, both for completeness and because we expect to observe similar results with the new outer bounds that we introduced in Chapter 5. 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

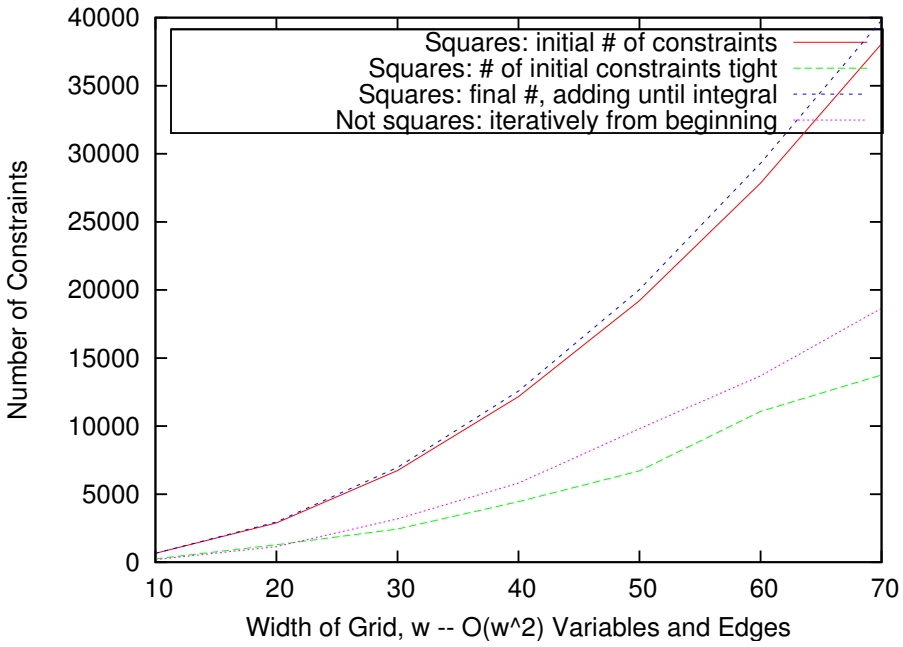


Figure 6-4: MAP on Ising grid graphs of width $w \times w$. On the y-axis we show the number of cycle inequalities that are added by the cutting-plane algorithm. We found the MAP solution in all trials.

Goemans-Williamson approximation algorithm give any such guarantee of optimality.

In Figure 6-4 we show results for MAP on Ising grid graphs with variables $x_i \in \{0, 1\}$. 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 LOCAL(G) alone were insufficient, giving fractional solutions for all trials. However, by 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). In 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 of 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(G) 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 (see "Squares: final" in figure), 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), in addition to trying to separate other classes of valid inequalities for the cut polytope.

In particular, 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.

Chapter 7

Conclusion

This thesis takes a new perspective on probabilistic inference, marrying variational inference algorithms with the cutting-plane methodology of combinatorial optimization and classical results from polyhedral combinatorics.

We show that using tighter outer bounds on the marginal polytope significantly improves the accuracy of predicting marginal probabilities in highly coupled MRFs. For the MRFs that we experiment with, the cutting-plane algorithm achieves these results with only a small number of additional inequalities. One reason 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. We hope 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.

Our work sheds some light on the relative value of the entropy approximation compared to the relaxation of the marginal polytope. When the MRF is weakly coupled, both the TRW and log-determinant entropy approximations do reasonably well using the local consistency polytope. This is not surprising: the limit of weak coupling is a fully disconnected graph, for which both the entropy approximation and the marginal polytope relaxation are exact. With the local consistency polytope, both entropy approximations get steadily worse as the coupling increases. In contrast, using the exact marginal polytope, we see a peak at $\theta = 2$, then a steady improvement as the coupling term grows. This occurs because the limit of strong coupling is the

MAP problem, for which using the exact marginal polytope will give exact results. The interesting region is near the peak, where the entropy term is neither exact nor outweighed by the coupling. Our algorithms seem to "solve" the part of the problem caused by the local consistency polytope relaxation, giving nearly as good results as the exact marginal polytope: TRW's accuracy goes from .33 to .15, and log-determinant's accuracy from .17 to .076. Regardless, the fact that neither entropy approximation can achieve accuracy below .07, even with the exact marginal polytope, motivates further research on improving this part of the approximation.

There are various directions to proceed, further strengthening the connection to polyhedral combinatorics. For example, many recent MRFs in vision and computational biology have matching constraints enforcing that two or more variables cannot be assigned the same value. While these constraints are usually imposed within the potentials, in the variational framework they correspond to taking the intersection of the usual marginal polytope with the matching polytope for the corresponding graph. For bipartite graphs, a linear number of constraints suffice to characterize the matching polytope, and these can be used to give a tighter outer bound on the marginal polytope. For general graphs, an efficient separation algorithm exists for the matching polytope using Gomory-Hu cut trees. These constraints can be directly used by our cutting-plane algorithm for MRFs with matching potentials.

The results in this thesis lead to several interesting open problems. The first is to get a better understanding of the new outer bound on the marginal polytope. Which of the inequalities obtained through projection are facet-defining for the marginal polytope? Does considering all possible partition schemes, given by the full projection graph, give strictly tighter relaxations than with a subset of the partition schemes such as the k-projection graph? The second set of questions are algorithmic. Can we bound the number of inequalities added for certain classes of MRFs? How can we project the odd-wheel and bicycle odd-wheel inequalities to yield an efficient algorithm for sparse graphs? Can we obtain fast separation heuristics using approximation algorithms? Finally, can we develop new message-passing algorithms which can incorporate cycle (and other) inequalities, to use them as an efficient inner loop in the cutting-plane

algorithm? Results for any of these problems would directly lead to new algorithms for both probabilistic inference and combinatorial optimization.

Appendix A

Remarks on Complexity

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

Theorem 6. *The following decision problem is NP-complete: given a vector* $\mu \in \mathbb{R}_+^{V_n \cup E_n}$*, 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 [11]. □

Bibliography

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[2] F. Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 60:53–68, 1993.

[3] F. Barahona, M. Grötschel, M. Junger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

[4] F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

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

[6] R. Borndörfer and R. Weismantel. Set packing relaxations of some integer programs. *Mathematical Programming*, 88:425–450, 2000. ZIB Report 97-30.

[7] R. Borndörfer and R. Weismantel. Discrete relaxations of combinatorial programs. *Discrete Applied Mathematics*, 112(1–3):11–26, 2001. ZIB Report 97-54.

[8] S. Chopra and J. Owen. Extended formulations of the A-cut problem. *Mathematical Programming*, 73:7–30, 1996.

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

[10] M. Deza, M. Grötschel, and M. Laurent. Clique-web facets for multicut polytopes. *Math. Oper. Res.*, 17(4):981–1000, 1992.

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

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

[13] K. Krishnan and J. E. Mitchell. A semidefinite programming based polyhedral cut and price approach for the maxcut problem. *Comput. Optim. Appl.*, 33(1):51–71, 2006.

[14] M. Leisink and B. Kappen. Bound propagation. *Journal of Artificial Intelligence Research*, 19:139–154, 2003.

[15] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[16] D. Sontag, R. Singh, and B. Berger. Probabilistic modeling of systematic errors in two-hybrid experiments. *Pacific Symposium on Biocomputing*, 12:445–457, 2007.

[17] M. Wainwright, T. Jaakkola, and A. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, July 2005.

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

[19] M. Wainwright and M. I. Jordan. Log-determinant relaxation for approximate inference in discrete Markov random fields. *IEEE Transactions on Signal Processing*, 54(6):2099–2109, June 2006.

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

[21] 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*, 7:1887–1907, September 2006.

[22] J. Yedidia, W. Freeman, and Y. Weiss. Bethe free energy, Kikuchi approximations, and belief propagation algorithms. Technical Report 16, Mitsubishi Electric Research Lab, 2001.

[23] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282– 2312, July 2005.