Exact MAP Inference by Avoiding Fractional Vertices

Erik M. Lindgren [1] **Alexandros G. Dimakis** [1] **Adam Klivans** [2]

Abstract

Given a graphical model, one essential problem is MAP inference, that is, finding the most likely configuration of states according to the model. Although this problem is NP-hard, large instances can be solved in practice and it is a major open question is to explain why this is true. We give a natural condition under which we can provably perform MAP inference in polynomial time—we require that the number of fractional vertices in the LP relaxation exceeding the optimal solution is bounded by a polynomial in the problem size. This resolves an open question by Dimakis, Gohari, and Wainwright. In contrast, for general LP relaxations of integer programs, known techniques can only handle a constant number of fractional vertices whose value exceeds the optimal solution. We experimentally verify this condition and demonstrate how efficient various integer programming methods are at removing fractional solutions.

1. Introduction

Given a graphical model, one essential problem is MAP inference, that is, finding the most likely configuration of states according to the model.

Consider graphical models with binary random variables and pairwise interactions, also known as Ising models. For a graph $G = (V, E)$ with node weights $\theta \in \mathbb{R}^V$ and edge weights $W \in \mathbb{R}^E$, the probability of a variable configuration is given by

$$\mathrm{P}(X = x) = \frac{1}{Z} \exp\left(\sum_{i \in V} \theta_i x_i + \sum_{ij \in E} W_{ij} x_i x_j \right), \quad (1)$$

where Z is a normalization constant.

The MAP problem is to find the configuration $x \in \{0,1\}^V$ that maximizes Equation (1). We can write this as an integer linear program (ILP) as follows:

$$\begin{aligned} \max_{q \in \mathbb{R}^{V \cup E}} \quad & \sum_{i \in V} \theta_i q_i + \sum_{ij \in E} W_{ij} q_{ij} \\ \text{s.t.} \quad & q_i \in \{0,1\} \quad \forall i \in V \\ & q_{ij} \geq \max\{0, q_i + q_j - 1\} \quad \forall ij \in E \\ & q_{ij} \leq \min\{q_i, q_j\} \quad \forall ij \in E. \end{aligned} \quad (2)$$

The MAP problem on binary, pairwise graphical models contains, as a special case, the Max-cut problem and is therefore NP-hard. For this reason, a significant amount of attention has focused on analyzing the *LP relaxation* of the ILP, which can be solved efficiently in practice.

$$\begin{aligned} \max_{q \in \mathbb{R}^{V \cup E}} \quad & \sum_{i \in V} \theta_i q_i + \sum_{ij \in E} W_{ij} q_{ij} \\ \text{s.t.} \quad & 0 \leq q_i \leq 1 \quad \forall i \in V \\ & q_{ij} \geq \max\{0, q_i + q_j - 1\} \quad \forall ij \in E \\ & q_{ij} \leq \min\{q_i, q_j\} \quad \forall ij \in E \end{aligned} \quad (3)$$

This relaxation has been an area of intense research in machine learning and statistics. In (Meshi et al., 2016), the authors state that a major open question is to identify why real world instances of Problem (2) can be solved efficiently despite the theoretical worst case complexity.

We make progress on this open problem by analyzing the *fractional vertices* of the LP relaxation, that is, the extreme points of the polytope with fractional coordinates. Vertices of the relaxed polytope with fractional coordinates are called pseudomarginals for graphical models and pseudocodewords in coding theory. If a fractional vertex has higher objective value (i.e. likelihood) compared to the best integral one, the LP relaxation fails. We call fractional vertices with an objective value at least as good as the objective

[1]Department of Electrical and Computer Engineering, University of Texas at Austin, USA [2]Department of Computer Science, University of Texas at Austin, USA. Correspondence to: Erik M. Lindgren <erikml@utexas.edu>, Alexandros G. Dimakis <dimakis@austin.utexas.edu>, Adam Klivans <klivans@cs.utexas.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

value of the optimal integral vertex **confounding vertices**. Our main result is that it is possible to prune all confounding vertices efficiently when their number is polynomial.

Our contributions:

- Our first contribution is a general result on integer programs. We show that *any* 0-1 integer linear program (ILP) can be solved exactly in polynomial time, if the number confounding vertices is bounded by a polynomial. This applies to MAP inference for a graphical model over any alphabet size and any order of connection. The same result (exact solution if the number of confounding vertices is bounded by a polynomial) was established by (Dimakis et al., 2009) for the special case of LP decoding of LDPC codes (Feldman et al., 2005). The algorithm from (Dimakis et al., 2009) relies on the special structure of the graphical models that correspond to LDPC codes. In this paper we generalize this result for any ILP in the unit hypercube. Our results extend to finding all integral vertices among the M-best vertices.

- Given our condition, one may be tempted to think that we generate the top M-best vertices of a linear program (for M polynomial) and output the best integral one in this list. We actually show that such an approach would be computationally intractable. Specifically, we show that it is NP-hard to produce a list of the M-best vertices if $M = O(n^{\varepsilon})$ for any fixed $\varepsilon > 0$. This result holds even if the list is allowed to be approximate. This strengthens the previously known hardness result (Angulo et al., 2014) which was $M = O(n)$ for the exact M-best vertices. In terms of achievability, the best previously known result (from (Angulo et al., 2014)) can only solve the ILP if there is at most a *constant* number of confounding vertices.

- We obtain a complete characterization of the fractional vertices of the local polytope for binary, pairwise graphical models. We show that any variable in the fractional support must be connected to a frustrated cycle by other fractional variables in the graphical model. This is a complete structural characterization that was not previously known, to the best of our knowledge.

- We develop an approach to estimate the number of confounding vertices of a half-integral polytope. We use this method in an empirical evaluation of the number of confounding vertices of previously studied problems and analyze how well common integer programming techniques perform at pruning confounding vertices.

2. Background and Related Work

For some classes of graphical models, it is possible to solve the MAP problem exactly. For example see (Weller et al., 2016) for balanced and almost balanced models, (Jebara, 2009) for perfect graphs, and (Wainwright et al., 2008) for graphs with constant tree-width.

These conditions are often not true in practice and a wide variety of general purpose algorithms are able to solve the MAP problem for large inputs. One class is belief propagation and its variants (Yedidia et al., 2000; Wainwright et al., 2003; Sontag et al., 2008). Another class involves general ILP optimization methods (see e.g. (Nemhauser & Wolsey, 1999)). Techniques specialized to graphical models include cutting-plane methods based on the cycle inequalities (Sontag & Jaakkola, 2007; Komodakis & Paragios, 2008; Sontag et al., 2012). See also (Kappes et al., 2013) for a comparative survey of techniques.

In (Weller et al., 2014), the authors investigate how pseudo-marginals and relaxations relate to the success of the Bethe approximation of the partition function.

There has been substantial prior work on improving inference building on these LP relaxations, especially for LDPC codes in the information theory community. This work ranges from very fast solvers that exploit the special structure of the polytope (Burshtein, 2009), connections to unequal error protection (Dimakis et al., 2007), and graphical model covers (Koetter et al., 2007). LP decoding currently provides the best known finite-length error-correction bounds for LDPC codes both for random (Daskalakis et al., 2008; Arora et al., 2009), and adversarial bit-flipping errors (Feldman et al., 2007).

For binary graphical models, there is a body of work which tries to exploit the *persistency* of the LP relaxation, that is, the property that integer components in the solution of the relaxation must take the same value in the optimal solution, under some regularity assumptions (Boros & Hammer, 2002; Rother et al., 2007; Fix et al., 2012).

Fast algorithms for solving large graphical models in practice include (Ihler et al., 2012; Dechter & Rish, 2003).

The work most closely related to this paper involves eliminating fractional vertices (so-called pseudocodewords in coding theory) by changing the polytope or the objective function (Zhang & Siegel, 2012; Chertkov & Stepanov, 2008; Liu et al., 2012).

3. Provable Integer Programming

A binary integer linear program is an optimization problem of the following form.

$$\begin{aligned} \max_x \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

which is relaxed to a linear program by replacing the $x \in \{0,1\}^n$ constraint with $0 \leq x \leq 1$. For binary integer programs with the box constraints $0 \leq x_i \leq 1$ for all i, every integral vector x is a *vertex* of the polytope described by the constraints of the LP relaxation. However *fraction vertices* may also be in this polytope, and fractional solutions can potentially have an objective value larger than every integral vertex.

If the optimal solution to the linear program happens to be integral, then this is the optimal solution to the original integer linear program. If the optimal solution is fractional, then a variety of techniques are available to *tighten* the LP relaxation and eliminate the fractional solution.

We establish a success condition for integer programming based on the number of confounding vertices, which to the best of our knowledge was unknown. The algorithm used in proving Theorem 1 is a version of branch-and-bound, a classic technique in integer programming (Land & Doig, 1960) (see (Nemhauser & Wolsey, 1999) for a modern reference on integer programming). This algorithm works by starting with a root node, then *branching* on a fractional coordinate by making two new linear programs with all the constraints of the parent node, with the constraint $x_i = 0$ added to one new *leaf* and $x_i = 1$ added to the other. The decision on which leaf of the tree to branch on next is based on which leaf has the best objective value. When the best leaf is integral, we know that this is the best integral solution. This algorithm is formally written in Algorithm 1.

Theorem 1. *Let x^* be the optimal integral solution and let $\{v_1, v_2, \ldots, v_M\}$ be the set of confounding vertices in the LP relaxation. Algorithm 1 will find the optimal integral solution x^* after $2M$ calls to an LP solver.*

Since MAP inference is a binary integer program regardless of the alphabet size of the variables and order of the clique potentials, we have the following corollary:

Corollary 2. *Given a graphical model such that the local polytope has M as cofounding variables, Algorithm 1 can find the optimal MAP configuration with $2M$ calls to an LP solver.*

Cutting-plane methods, which remove a fractional vertex by introducing a new constraint in the polytope may not have this property, since this cut may create new confounding vertices. This branch-and-bound algorithm has the desirable property that it never creates a new fractional vertex. We note that other branching algorithms, such as the algorithm presented by the authors in (Marinescu & Dechter, 2009), do not immediately allow us to prove our desired theorem.

Note that *warm starting* a linear program with slightly modified constraints allows subsequent calls to an LP solver to be much more efficient after the root LP has been solved.

Algorithm 1 Branch and Bound

test
Input: an LP $\{\min c^T x : Ax \leq b, 0 \leq x \leq 1\}$

\# branch (v, I_0, I_1) means v is optimal LP
\# with $x_{I_0} = 0$ and $x_{I_1} = 1$.
def LP(I_0, I_1):
  $v* \leftarrow \arg\max c^T x$
  subject to:
    $Ax \leq b$
    $x_{I_0} = 0$
    $x_{I_1} = 1$
  return v^* if feasible, else return null

$v \leftarrow \text{LP}(\emptyset, \emptyset)$
$B \leftarrow \{(v, \emptyset, \emptyset)\}$
while optimal integral vertex not found:
  $(v, I_0, I_1) \leftarrow \arg\max_{(v,I_0,I_1)\in B} c^T v$
  if v is integral:
    return v
  else:
    find a fractional coordinate i
    $v^{(0)} \leftarrow \text{LP}(I_0 \cup \{i\}, I_1)$
    $v^{(1)} \leftarrow \text{LP}(I_0, I_1 \cup \{i\})$
    remove (v, I_0, I_1) from B
    add $(v^{(0)}, I_0 \cup \{i\}, I_1)$ to B if feasible
    add $(v^{(1)}, I_0, I_1 \cup \{i\})$ to B if feasible

3.1. Proof of Theorem 1

The proof follows from the following invariants:

- At every iteration we remove at least one fractional vertex.
- Every integral vertex is in exactly one branch.
- Every fractional vertex is in at most one branch.
- No fractional vertices are created by the new constraints.

To see the last invariant, note that every vertex of a polytope can be identified by the set of inequality constraints that are satisfied with equality (see (Bertsimas & Tsitsiklis, 1997)). By forcing an inequality constraint to be tight, we cannot possibly introduce new vertices.

3.2. The M-Best LP Problem

As mentioned in the introduction, the algorithm used to prove Theorem 1 does not enumerate all the fractional vertices until it finds an integral vertex. Enumerating the M-best vertices of a linear program is the M-best LP problem.

Definition. Given a linear program $\{\min c^T x : x \in P\}$ over a polytope P and a positive integer M, the *M-best LP problem* is to optimize

$$\max_{\{v_1,\ldots,v_M\}\subseteq V(P)} \sum_{k=1}^{M} c^T v_k.$$

This was established by (Angulo et al., 2014) to be NP-hard when $M = O(n)$. We strengthen this result to hardness of approximation even when $M = n^\varepsilon$ for any $\varepsilon > 0$.

Theorem 3. *It is NP-hard to approximate the M-best LP problem by a factor better than $O(\frac{n^\varepsilon}{M})$ for any fixed $\varepsilon > 0$.*

Proof. Consider the circulation polytope described in (Khachiyan et al., 2008), with the graph and weight vector w described in (Boros et al., 2011). By adding an $O(\log M)$ long series of 2×2 bipartite subgraphs, we can make it such that one long path in the original graph implies M long paths in the new graph, and thus it is NP-hard to find any of these long paths in the new graph. By adding the constraint vector $w^T x \leq 0$, and using the cost function $-w$, the vertices corresponding to the short paths have value $1/2$, the vertices corresponding to the long paths have value $O(1/n)$, and all other vertices have value 0. Thus the optimal set has value $O(n + \frac{M}{n})$. However it is NP-hard to find a set of value greater than $O(n)$ in polynomial time, which gives an $O(\frac{n}{M})$ approximation. Using a padding argument, we can replace n with n^ε. □

The best known algorithm for the M-best LP problem is a generalization of the facet guessing algorithm (Dimakis et al., 2009) developed in (Angulo et al., 2014), which would require $O(m^M)$ calls to an LP solver, where m is the number of constraints of the LP. Since we only care about integral solutions, we can find the single best integral vertex with $O(M)$ calls to an LP solver, and if we want all of the K-best integral solutions among the top M vertices of the polytope, we can find these with $O(nK + M)$ calls to an LP-solver, as we will see in the next section.

3.3. K-Best Integral Solutions

Finding the K-best solutions to general optimization problems has been uses in several machine learning applications. Producing multiple high-value outputs can be naturally combined with post-processing algorithms that select the most desired solution using additional side-information. There is a significant volume of work in the general area, see (Fromer & Globerson, 2009; Batra et al., 2012) for MAP solutions in graphical models and (Eppstein, 2014) for a survey on M-best problems.

We further generalize our theorem to find the K-best integral solutions.

Theorem 4. *Under the assumption that there are less than M fractional vertices with objective value at least as good as the K-best integral solutions, we can find all of the K-best integral solutions, $O(nK + M)$ calls to an LP solver.*

The algorithm used in this theorem is Algorithm 2. It combines Algorithm 1 with the space partitioning technique used in (Murty, 1968; Lawler, 1972). If the current optimal solution in the solution tree is fractional, then we use the branching technique in Algorithm 1. If the current optimal solution in the solution tree x^* is integral, then we branch by creating a new leaf for every i not currently constrained by the parent with the constraint $x_i = \neg x_i^*$.

4. Fractional Vertices of the Local Polytope

We now describe the fractional vertices of the local polytope for binary, pairwise graphical models, which is described in Equation 3. It was shown in (Padberg, 1989) that all the vertices of this polytope are *half-integral*, that is, all coordinates have a value from $\{0, \frac{1}{2}, 1\}$ (see (Weller et al., 2016) for a new proof of this).

Given a half-integral point $q \in \{0, \frac{1}{2}, 1\}^{V \cup E}$ in the local polytope, we say that a cycle $C = (V_C, E_C) \subseteq G$ is *frustrated* if there is an odd number of edges $ij \in E_C$ such that $q_{ij} = 0$. If a point q has a frustrated cycle, then it is a *pseudomarginal*, as no probability distribution exists that has as its singleton and pairwise marginals the coordinates of q. Half-integral points q with a frustrated cycle do not satisfy the *cycle inequalities* (Sontag & Jaakkola, 2007; Wainwright et al., 2008), for all cycles $C = (V_C, E_C), F = (V_F, E_F) \subseteq C, |E_F|$ odd we must have

$$\sum_{ij\in E_F} q_i + q_j - 2q_{ij} - \sum_{ij\in E_C\setminus E_F} q_i + q_j - 2q_{ij} \leq |F_C| - 1. \quad (4)$$

Frustrated cycles allow a solution to be zero on negative weights in a way that is not possible for any integral solution.

We have the following theorem describing all the vertices of the local polytope for binary, pairwise graphical models.

Algorithm 2 M-best Integral

```
Input: an LP {max c^T x : Ax ≤ b, 0 ≤ x ≤ 1}
Input: number of solutions K

def LP(I_0, I_1): same as Algorithm 1

def SplitIntegral(v, I_0, I_1):
  P ← { }
  for i ∈ [n] if i ∉ I_0 ∪ I_1:
    a ← ¬v_i
    I'_0, I'_1 ← copy(I_0, I_1)
    add i to I'_a
    v' ← LP(I'_0, I'_1)
    add (v', I'_0, I'_a) to P if feasible
  return P

v ← LP(∅, ∅)
B ← {(v, ∅, ∅)}
results ← { }
while K integral vertices not found:
  (v, I_0, I_1) ← arg max_{(v,I_0,I_1)∈B} c^T v
  if v is integral:
    add v to results
    add SplitIntegeral(v, I_0, I_1) to B
    remove (v, I_0, I_1) from B
  else:
    find a fractional coordinate i
    v^(0) ← LP(I_0 ∪ {i}, I_1)
    v^(1) ← LP(I_0, I_1 ∪ {i})
    remove (v, I_0, I_1) from B
    add (v^(0), I_0 ∪ {i}, I_1) to B if feasible
    add (v^(1), I_0, I_1 ∪ {i}) to B if feasible
return results
```

Theorem 5. *Given a point q in the local polytope, q is a vertex of this polytope if and only if $q \in \{0, \frac{1}{2}, 1\}^{V \cup E}$ and the induced subgraph on the fractional nodes of q is such that every connected component of this subgraph contains a frustrated cycle.*

4.1. Proof of Theorem 5

Every vertex q of an n-dimensional polytope is such that there are n constraints such that q satisfies them with equality, known as *active constraints* (see (Bertsimas & Tsitsiklis, 1997)). Every integral q is thus a vertex of the local polytope. We now describe the fractional vertices of the local polytope.

Definition. Let $q \in \{0, \frac{1}{2}, 1\}^{n+m}$ be a point of the local polytope. Let $G_F = (V_F, E_F)$ be an induced subgraph of points such that $q_i = \frac{1}{2}$ for all $i \in V_F$. We say that G_F is *full rank* if the following system of equations is full rank.

$$
\begin{aligned}
q_i + q_j - q_{ij} &= 1 \quad \forall ij \in E_F \text{ such that } q_{ij} = 0 \\
q_{ij} &= 0 \quad \forall ij \in E_F \text{ such that } q_{ij} = 0 \\
q_i - q_{ij} &= 0 \quad \forall ij \in E_F \text{ such that } q_{ij} = \frac{1}{2} \\
q_j - q_{ij} &= 0 \quad \forall ij \in E_F \text{ such that } q_{ij} = \frac{1}{2}
\end{aligned}
\tag{5}
$$

Theorem 5 follows from the following lemmas.

Lemma 6. *Let $q \in \{0, \frac{1}{2}, 1\}^{n+m}$ be a point of the local polytope. Let $G_F = (V_F, E_F)$ be the subgraph induced by the nodes $i \in V$ such that $q_i = \frac{1}{2}$. The point q is a vertex if and only if every connected component of G_F is full rank.*

Lemma 7. *Let $q \in \{0, \frac{1}{2}, 1\}^{n+m}$ be a point of the local polytope. Let $G_F = (V_F, E_F)$ be a connected subgraph induced from nodes that such that such that $q_i = \frac{1}{2}$ for all $i \in V_F$. G_F is full rank if and only if G_F contains cycle that is full rank.*

Lemma 8. *Let $q \in \{0, \frac{1}{2}, 1\}^{n+m}$ be a point of the local polytope. Let $C = (V_C, E_C)$ be a cycle of G such that q_i is fractional for all $i \in V_C$. C is full rank if and only if C is a frustrated cycle.*

Proof of Lemma 6. Suppose every connected component is full rank. Then every fractional node and edge between fractional nodes is fully specified by their corresponding equations in Problem 3. It is easy to check that all integral nodes, edges between integral nodes, and edges between integral and fractional nodes is also fixed. Thus q is a vertex.

Now suppose that there exists a connected component that is not full rank. The only other constraints involving this connected component are those between fractional nodes and integral nodes. However, note that these constraints are always rank 1, and also introduce a new edge variable. Thus all the constraints where q is tight do not make a full rank system of equations. □

Proof of Lemma 7. Suppose G_F has a full rank cycle. We will build the graph starting with the full rank cycle then adding one connected edge at a time. It is easy to see from Equations 5 that all new variables introduced to the system of equations have a fixed value, and thus the whole connected component is full rank.

Now suppose G_F has no full rank cycle. We will again build the graph starting from the cycle then adding one connected edge at a time. If we add an edge that connects to a new node, then we added two variables and two equations, thus we did not make the graph full rank. If we add an edge between two existing nodes, then we have a cycle involving this edge. We introduce two new equations, however with one of the equations and the other cycle equations, we can produce the other equation, thus we can increase the rank by one but we also introduced a new edge. Thus the whole graph cannot be full rank. □

The proof of Lemma 8 from the following lemma.

Lemma 9. *Consider a collection of n vectors*

$$\begin{aligned} v_1 &= (1, t_1, 0, \ldots, 0) \\ v_2 &= (0, 1, t_2, 0, \ldots, 0) \\ v_3 &= (0, 0, 1, t_3, 0, \ldots, 0) \\ &\vdots \\ v_{n-1} &= (0, \ldots, 0, 1, t_{n-1}) \\ v_n &= (t_n, 0, \ldots, 0, 1) \end{aligned}$$

for $t_i \in \{-1, 1\}$. We have $\operatorname{rank}(v_1, v_2, \ldots, v_n) = n$ if and only if there is an odd number of vectors such that $t_i = 1$.

Proof of Lemma 9. Let k be the number of vectors such that $t_i = 1$. Let $S_1 = v_1$ and define

$$S_{i+1} = \begin{cases} S_i - v_{i+1} & \text{if } S_i(i+1) = 1 \\ S_i + v_{i+1} & \text{if } S_i(i+1) = -1 \end{cases}$$

for $i = 2, \ldots, n-1$.

Note that if $t_{i+1} = -1$ then $S_{i+1}(i+2) = S_i(i+1)$ and if $t_{i+1} = 1$ then $S_{i+1}(i+2) = -S_i(i+1)$. Thus the number of times the sign changes is exactly the number of $t_i = 1$ for $i \in \{2, \ldots, n-1\}$.

Using the value of S_{n-1} we can now we can check for all values of t_1 and t_n that the following is true.

- If k is odd then $(1, 0, \ldots, 0) \in \operatorname{span}(v_1, v_2, \ldots, v_n)$, which allows us to create the entire standard basis, showing the vectors are full rank.
- If k is even then $v_n \in \operatorname{span}(v_1, v_2, \ldots, v_{n-1})$ and thus the vectors are not full rank.

□

5. Estimating the number of Confounding Singleton Marginals

For this section we generalize generalize Theorem 1. We see after every iteration we potentially remove more than one confounding vertex—we remove all confounding vertices that agree with $x_{I_0} = 0$ and $x_{I_1} = 1$ and are fractional with any value at coordinate i. We also observe that we can handle a mixed integer program (MIP) with the same algorithm.

$$\begin{aligned} \max_x \quad & c^T x + d^T z \\ \text{subject to} \quad & Ax + Bz \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

We will call a vertex (x, z) fractional if its x component is fractional. For each fractional vertex (x, z), we create a half-integral vector $S(x)$ such that

$$S(x)_i = \begin{cases} 0 & \text{if } x_i = 0 \\ \frac{1}{2} & \text{if } x_i \text{ is fractional} \\ 1 & \text{if } x_i = 1 \end{cases}$$

For a set of vertices V, we define $S(V)$ to be the set $\{S(x) : (x, z) \in V\}$, i.e. we remove all duplicate entries.

Theorem 10. *Let (x^*, z^*) be the optimal integral solution and let V_C be the set of confounding vertices. Algorithm 1 will find the optimal integral solution (x^*, z^*) after $2|S(V_C)|$ calls to an LP solver.*

For MAP inference in graphical models, $S(V_C)$ refers to the fractional singleton marginals q_V such that there exists a set of pairwise pseudomarginals q_E such that (q_V, q_E) is a cofounding vertex. In this case we call q_V a confounding singleton marginal. We develop Algorithm 3 to estimate the number of confounding singleton marginals for our experiments section. It is based on the k-best enumeration method developed in (Murty, 1968; Lawler, 1972).

Algorithm 3 works by a branching argument. The root node is the original LP. A leaf node is branched on by introducing a new leaf for every node in V and every element of $\{0, \frac{1}{2}, 1\}$ such that $q_i \neq a$ in the parent node and the constraint $\{q_i = a\}$ is not in the constraints for the parent node. For $i \in V$, $a \in \{0, \frac{1}{2}, 1\}$, we create the leaf such that it has all the constraints of its parents plus the constraint $q_i = a$.

Note that Algorithm 3 actually generates a superset of the elements of $S(V_C)$, since the introduction of constraints of the type $q_i = \frac{1}{2}$ introduce vertices into the new polytope that were not in the original polytope. This does not seem to be an issue for the experiments we consider, however this does occur for other graphs. An interesting question is if the vertices of the local polytope can be provably enumerated.

6. Experiments

We consider a synthetic experiment on randomly created graphical models, which were also used in (Sontag & Jaakkola, 2007; Weller, 2016; Weller et al., 2014). The graph topology used is the complete graph on 12 nodes. We first reparametrize the model to use the sufficient statistics

Algorithm 3 Estimate $S(V_C)$ for Binary, Pairwise Graphical Models

Input: a binary, pairwise graphical model LP

# branch $(v, I_0, I_{\frac{1}{2}}, I_1)$ means v is optimal LP
# with $x_{I_0} = 0, x_{I_{\frac{1}{2}}} = \frac{1}{2}$, and $x_{I_1} = 1$.
def $\mathrm{LP}(I_0, I_{\frac{1}{2}}, I_1)$:
  optimize LP with additional constraints:
    $x_{I_0} = 0$
    $x_{I_{\frac{1}{2}}} = \frac{1}{2}$
    $x_{I_1} = 1$
  return q^* if feasible, else return null

$q \leftarrow \mathrm{LP}(\emptyset, \emptyset, \emptyset)$
$B \leftarrow \{(q, \emptyset, \emptyset, \emptyset)\}$
solution $\leftarrow \{\ \}$
while optimal integral vertex not found:
  $(q, I_0, I_{\frac{1}{2}}, I_1) \leftarrow \arg\max_{(q, I_0, I_{\frac{1}{2}}, I_1) \in B}$ objective val
  add q to solution
  remove $(q, I_0, I_{\frac{1}{2}}, I_1)$ from B
  for $i \in V$ if $i \notin I_0 \cup I_{\frac{1}{2}} \cup I_1$:
    for $a \in \{0, \frac{1}{2}, 1\}$ if $q_i \neq a$:
      $I_0', I_{\frac{1}{2}}', I_1' \leftarrow \mathrm{copy}(I_0, I_{\frac{1}{2}}, I_1)$
      $I_a' \leftarrow I_a' \cup \{i\}$
      $q' \leftarrow \mathrm{LP}(I_0', I_{\frac{1}{2}}', I_1')$
      add $(q', I_0', I_{\frac{1}{2}}', I_1')$ to B if feasible
return solution

$\mathbb{1}(x_i = x_j)$ and $\mathbb{1}(x_i = 1)$. The node weights are drawn $\theta_i \sim \mathrm{Uniform}(-1, 1)$ and the edge weights are drawn $W_{ij} \sim \mathrm{Uniform}(-w, w)$ for varying w. The quantity w determines how strong the connections are between nodes. We do 100 draws for each choice of edge strength w.

For the complete graph, we observe that Algorithm 3 does not yield any points that do not correspond to vertices, however this does occur for other topologies.

We first compare how the number of fractional singleton marginals $|S(V_C)|$ changes with the connection strength w. In Figure 1, we plot the sample CDF of the probability that $|S(V_C)|$ is some given value. We observe that $|S(V_C)|$ increases as the connection strength increases. Further we see that while most instances have a small number for $|S(V_C)|$, there are rare instances where $|S(V_C)|$ is quite large.

Now we compare how the number of cycle constraints from Equation (4) that need to be introduced to find the best integral solution changes with the number of confounding singleton marginals in Figure 2. We use the algorithm for finding the most frustrated cycle in (Sontag & Jaakkola, 2007) to introduce new constraints. We observe that each constraint seems to remove many confounding singleton marginals.

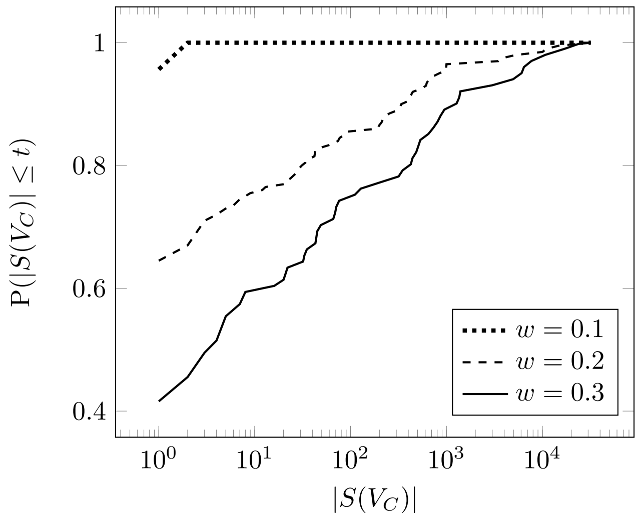

Figure 1. We compare how the number of fractional singleton marginals $|S(V_C)|$ changes with the connection strength w. We plot the sample CDF of the probability that $|S(V_C)|$ is some given value. We observe that $|S(V_C)|$ increases as the connection strength increases. Further we see that while most instances have a small number for $|S(V_C)|$, there are rare instances where $|S(V_C)|$ is quite large.

We also observe the number of introduced confounding singleton marginals that are introduced by the cycle constraints increases with the number of confounding singleton marginals in Figure 3.

Finally we compare the number of branches needed to find the optimal solution increases with the number of confounding singleton marginals in Figure 4. A similar trend arises as with the number of cycle inequalities introduced. To compare the methods, note that branch-and-bound uses twice as many LP calls as there are branches. For this family of graphical models, branch-and-bound tends to require less calls to an LP solver than the cut constraints.

7. Conclusion

Perhaps the most interesting follow-up question to our work is to determine when, in theory and practice, our condition on the number of confounding pseudomarginals in the LP relaxation is small. Another interesting question is to see if it is possible to prune the number of confounding pseudomarginals at a faster rate. The algorithm presented for our main theorem removes one pseudomarginal after two calls to an LP solver. Is it possible to do this at a faster rate? From our experiments, this seems to be the case in practice.

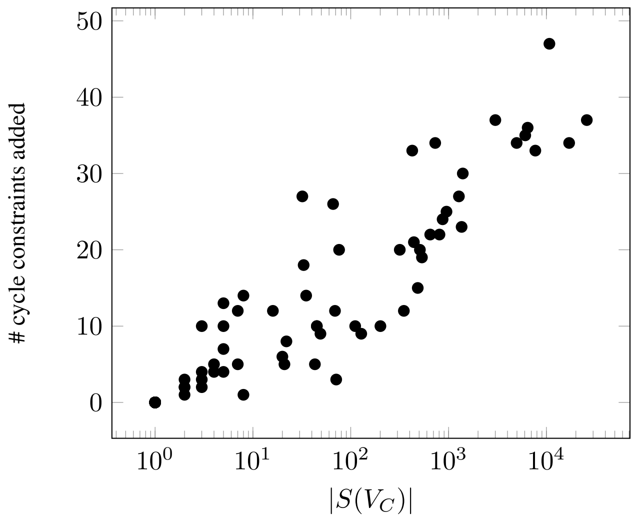

Figure 2. We compare how the number of cycle constraints from Equation (4) that need to be introduced to find the best integral solution changes with the number of confounding singleton marginals. We use the algorithm for finding the most frustrated cycle in (Sontag & Jaakkola, 2007) to introduce new constraints. We observe that each constraint seems to remove many confounding singleton marginals.

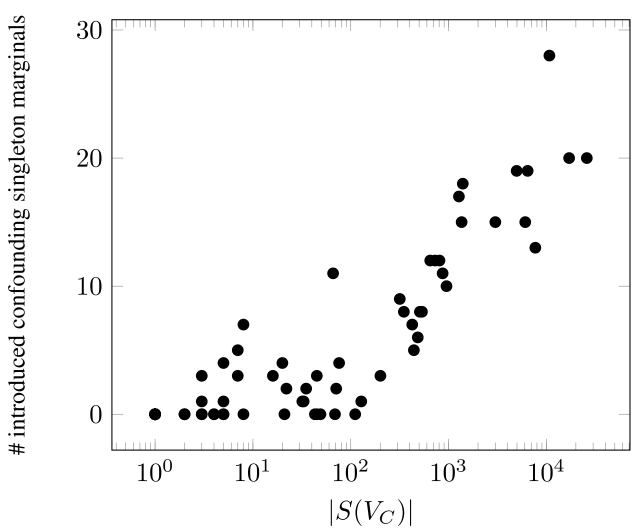

Figure 3. We also observe the number of introduced confounding singleton marginals that are introduced by the cycle constraints increases with the number of confounding singleton marginals .

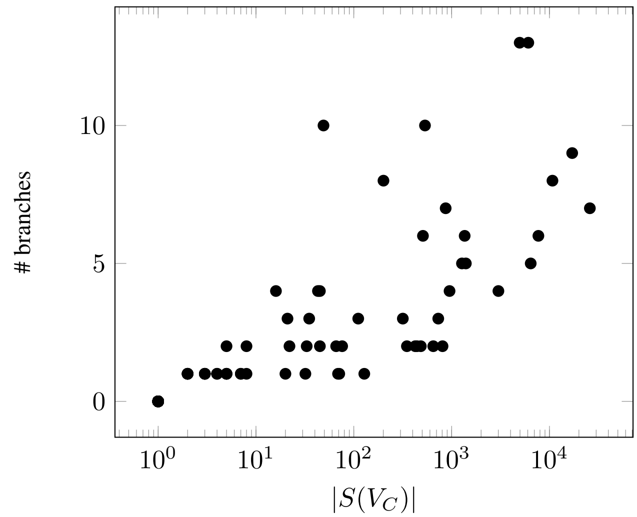

Figure 4. Finally we compare the number of branches needed to find the optimal solution increases with the number of confounding singleton marginals in Figure 4. A similar trend arises as with the number of cycle inequalities introduced. To compare the methods, note that branch-and-bound uses twice as many LP calls as there are branches. For this family of graphical models, branch-and-bound tends to require less calls to an LP solver than the cut constraints.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1110007 as well as NSF Grants CCF 1344364, 1407278, 1422549, 1618689, 1018829 and ARO YIP W911NF-14-1-0258.

References

Angulo, Gustavo, Ahmed, Shabbir, Dey, Santanu S, and Kaibel, Volker. Forbidden vertices. *Mathematics of Operations Research*, 40(2):350–360, 2014.

Arora, Sanjeev, Daskalakis, Constantinos, and Steurer, David. Message passing algorithms and improved lp decoding. In *Symposium on Theory of Computing*, pp. 3–12. ACM, 2009.

Batra, Dhruv, Yadollahpour, Payman, Guzman-Rivera, Abner, and Shakhnarovich, Gregory. Diverse m-best solutions in markov random fields. In *European Conference on Computer Vision*, pp. 1–16. Springer, 2012.

Bertsimas, Dimitris and Tsitsiklis, John N. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

Boros, Endre and Hammer, Peter L. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1):155–225, 2002.

Boros, Endre, Elbassioni, Khaled, Gurvich, Vladimir, and Tiwary, Hans Raj. The negative cycles polyhedron and hardness of checking some polyhedral properties. *Annals of Operations Research*, 188(1):63–76, 2011.

Burshtein, David. Iterative approximate linear programming decoding of LDPC codes with linear complexity. *IEEE Transactions on Information Theory*, 55(11): 4835–4859, 2009.

Chertkov, Michael and Stepanov, Mikhail G. An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes. *IEEE Transactions on Information Theory*, 54(4):1514–1520, 2008.

Daskalakis, Constantinos, Dimakis, Alexandros G, Karp, Richard M, and Wainwright, Martin J. Probabilistic analysis of linear programming decoding. *IEEE Transactions on Information Theory*, 54(8):3565–3578, 2008.

Dechter, Rina and Rish, Irina. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM (JACM)*, 50(2):107–153, 2003.

Dimakis, Alexandros G, Wang, Jiajun, and Ramchandran, Kannan. Unequal growth codes: Intermediate performance and unequal error protection for video streaming. In *Multimedia Signal Processing*, pp. 107–110. IEEE, 2007.

Dimakis, Alexandros G, Gohari, Amin A, and Wainwright, Martin J. Guessing facets: Polytope structure and improved LP decoder. *IEEE Transactions on Information Theory*, 55(8):3479–3487, 2009.

Eppstein, D. k-best enumeration. *Encyclopedia of Algorithms*, 2014.

Feldman, Jon, Wainwright, Martin J, and Karger, David R. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51(3):954–972, 2005.

Feldman, Jon, Malkin, Tal, Servedio, Rocco A, Stein, Cliff, and Wainwright, Martin J. LP decoding corrects a constant fraction of errors. *IEEE Transactions on Information Theory*, 53(1):82–89, 2007.

Fix, Alexander, Chen, Joyce, Boros, Endre, and Zabih, Ramin. Approximate MRF inference using bounded treewidth subgraphs. *Computer Vision–ECCV 2012*, pp. 385–398, 2012.

Fromer, Menachem and Globerson, Amir. An LP view of the M-best MAP problem. In *NIPS*, volume 22, pp. 567–575, 2009.

Ihler, Alexander, Flerova, Natalia, Dechter, Rina, and Otten, Lars. Join-graph based cost-shifting schemes. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 397–406. AUAI Press, 2012.

Jebara, Tony. Map estimation, message passing, and perfect graphs. In *UAI*, pp. 258–267. AUAI Press, 2009.

Kappes, Joerg, Andres, Bjoern, Hamprecht, Fred, Schnorr, Christoph, Nowozin, Sebastian, Batra, Dhruv, Kim, Sungwoong, Kausler, Bernhard, Lellmann, Jan, Komodakis, Nikos, et al. A comparative study of modern inference techniques for discrete energy minimization problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1328–1335, 2013.

Khachiyan, Leonid, Boros, Endre, Borys, Konrad, Elbassioni, Khaled, and Gurvich, Vladimir. Generating all vertices of a polyhedron is hard. *Discrete & Computational Geometry*, 39(1-3):174–190, 2008.

Koetter, Ralf, Li, Wen-Ching W, Vontobel, Pascal O, and Walker, Judy L. Characterizations of pseudo-codewords of (low-density) parity-check codes. *Advances in Mathematics*, 213(1):205–229, 2007.

Komodakis, Nikos and Paragios, Nikos. Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *European Conference on Computer Vision*, pp. 806–820. Springer, 2008.

Land, Ailsa H and Doig, Alison G. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pp. 497–520, 1960.

Lawler, Eugene L. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management science*, 18(7):401–405, 1972.

Liu, Xishuo, Draper, Stark C, and Recht, Benjamin. Suppressing pseudocodewords by penalizing the objective of LP decoding. In *Information Theory Workshop*, pp. 367–371. IEEE, 2012.

Marinescu, Radu and Dechter, Rina. And/or branch-and-bound search for combinatorial optimization in graphical models. *Artificial Intelligence*, 173(16-17):1457–1491, 2009.

Meshi, Ofer, Mahdavi, Mehrdad, Weller, Adrian, and Sontag, David. Train and test tightness of LP relaxations in structured prediction. In *ICML*, 2016.

Murty, Katta G. Letter to the editoran algorithm for ranking all the assignments in order of increasing cost. *Operations research*, 16(3):682–687, 1968.

Nemhauser, George L and Wolsey, Laurence A. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1999.

Padberg, Manfred. The boolean quadric polytope: some characteristics, facets and relatives. *Mathematical programming*, 45(1):139–172, 1989.

Rother, Carsten, Kolmogorov, Vladimir, Lempitsky, Victor, and Szummer, Martin. Optimizing binary MRFs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, 2007.

Sontag, David and Jaakkola, Tommi S. New outer bounds on the marginal polytope. In *NIPS*, volume 20, pp. 1393–1400, 2007.

Sontag, David, Meltzer, Talya, Globerson, Amir, Jaakkola, Tommi, and Weiss, Yair. Tightening LP relaxations for map using message passing. In *UAI*, pp. 503–510. AUAI Press, 2008.

Sontag, David, Li, Yitao, et al. Efficiently searching for frustrated cycles in map inference. In *UAI*, 2012.

Wainwright, Martin J, Jaakkola, Tommi S, and Willsky, Alan S. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In *AISTATS*, 2003.

Wainwright, Martin J, Jordan, Michael I, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2): 1–305, 2008.

Weller, Adrian. Uprooting and rerooting graphical models. In *ICML*, 2016.

Weller, Adrian, Tang, Kui, Jebara, Tony, and Sontag, David. Understanding the bethe approximation: when and how can it go wrong? In *UAI*, pp. 868–877, 2014.

Weller, Adrian, Rowland, Mark, and Sontag, David. Tightness of LP relaxations for almost balanced models. In *AISTATS*, 2016.

Yedidia, Jonathan S, Freeman, William T, Weiss, Yair, et al. Generalized belief propagation. In *NIPS*, volume 13, pp. 689–695, 2000.

Zhang, Xiaojie and Siegel, Paul H. Adaptive cut generation algorithm for improved linear programming decoding of binary linear codes. *IEEE Transactions on Information Theory*, 58(10):6581–6594, 2012.