Facet Guessing for Finding the M-Best Integral Solutions of a Linear Program

Erik M. Lindgren* **Alexandros G. Dimakis*** **Adam Klivans†**
The University of Texas at Austin
*Department of Electrical and Computer Engineering †Department of Computer Science
erikml@utexas.edu dimakis@austin.utexas.edu klivans@cs.utexas.edu

Abstract

In machine learning, it is often advantageous to return multiple solutions to an optimization problem, rather than the single optimal solution. We consider the M-best LP problem, where the goal is to find the M vertices of a polytope that minimize a linear cost function. We present an algorithm that can find the M-best when M is constant. We complement this by a hardness of approximation result.

Furthermore, we study the problem of recovering the highest vertex with *integral coordinates*. This question arises naturally for MAP inference in graphical models. We show that for any polytope contained inside the unit hypercube, it is possible to find the *highest integral vertex* if it lies within the top $poly(n)$ vertices of the LP. This allows us to avoid polynomially many fractional vertices and still recover the optimal vertex with $\{0, 1\}$ coordinates. Our work resolves an open problem by Dimakis, Gohari, and Wainwright.

1 Introduction

Consider a linear program (LP) $\{\min c^T x : x \in P\}$ over a polytope P: the optimal solution of this LP is at the vertex[1] of P with the highest dot product with the vector $-c$. One can consider the whole list of the vertices of the polytope P, sorted by dot product with $-c$, and think that the LP solver is producing the top element in this list. In this paper we ask the question: is it possible to produce other vertices from this list, i.e. the second best, or the top M-best.

Finding the M-best solutions to general optimization problems has 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 [15] for structured prediction, [13, 3, 12] for MAP solutions in graphical models, [21] for sentence parsing in natural language processing, and [20] for pose estimation in computer vision. See also [9] for a survey on M-best problems.

Despite the significant volume of prior work, we are aware only of [7, 13, 1] that specifically consider linear programs.

Our results: We informally summarize our results. We assume that the polytope P is given as an H-description i.e., an intersection of polynomially many closed halfspaces (linear inequalities).

Upper Bounds: Generalizing work due to Dimakis, Gohari, and Wainwright [7], we introduce an algorithm called *generalized facet guessing* for finding the M-best vertices of an arbitrary bounded LP. Our algorithm runs in polynomial time for any $M = O(1)$. This result is also implicit in the work of [1], where it is stated in the language of forbidden vertices and extension complexity.

[1] We ignore the cases when a whole facet is optimal, for the first discussion.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

We build on these ideas and then consider the problem of finding the *highest integral vertex* if it lies within the top $\mathrm{poly}(n)$ vertices of a linear program. This has applications within coding theory and graphical models, where an integer program for MAP is relaxed to a linear program and unwanted fractional vertices may be introduced[2]. We solve this problem for general polytopes contained within the unit hypercube in a way that allows us to avoid polynomially many fractional vertices and still recover the optimal vertex with $\{0, 1\}$ coordinates. We then extend this to find the top M fractional vertices, under the assumption that they all lie within the top $\mathrm{poly}(n)$ vertices.

The only previous work for this problem we know of is due to Dimakis et al. [7] who showed that for the special-case inference problem of LDPC decoding, it is possible to find this highest integral coordinate vertex if it lies within $M = \mathrm{poly}(n)$ top list, with high probability. This required several assumptions about the structure of the graphical model and only applied to linear binary codes.

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 [4], connections to unequal error protection [8], and graphical model covers [17]. LP decoding currently provides the best known finite-length error-correction bounds for LDPC codes both for random [6, 2], and adversarial bit-flipping errors [10].

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 [24, 5, 18]. To the best of our knowledge, no prior method was known to succeed when polynomially many fractional vertices have higher likelihood over the best integral one. Our results imply that a specific version of branch and bound can always achieve that.

Hardness Results: 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 strengthens the previously known hardness result [1] which was $M = O(n)$. We extend these results to approximating the M-best vertices.

2 Definitions and Setup

A linear program is an optimization problem with a linear cost function constrained by an intersection of half spaces, which are constraints of the form $Ax \leq b$. If the constraint set is bounded, then the constraint set is a polytope P. The vertices of the polytope are the minimal set of points $V(P)$ such that the convex hull $\mathrm{conv}(V(P)) = P$. As is well known, the optimal solution to a bounded linear program always occurs at a vertex, and we can alway find an optimal vertex in polynomial time. Thus we define the M-best LP problem as the following:

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

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

The *max-M-best LP problem* is defined by changing the minimization with maximization.

For unbounded linear programs, it is NP-hard to find even the single best vertex [14], and so throughout we assume all linear programs are bounded. For many problems of interest, we can add constraints to make the problem bounded without changing the solution to the M-best problem.

3 M-Best LP

In [1], it was show that the M-best LP problem was NP-hard to solve even if M is $O(n)$. By a padding argument, we see that the problem remains NP-hard even if $M = O(n^{\varepsilon})$ for any fixed $\varepsilon > 0$. We extend this to hardness of approximation by establishing the following hardness of approximation results.

[2]This relaxed polytope is called the Fundamental polytope or Relaxed polytope for the special case of LDPC codes or the Local polytope in general, see e.g. [11, 22]

Theorem 1. *The min- and max-M-best LP problems are hard to approximate in the following senses:*

- *It is NP-hard to approximate the total cost in the min-M-best LP problem of a bounded linear program within a factor $O(2^{n^c})$, even if M is restricted to be $O(n^\varepsilon)$ for any fixed c and $\varepsilon > 0$.*
- *It is NP-hard to approximate the max-M-best LP problem by a factor better than $O(\frac{n^\varepsilon}{M})$ for any fixed $\varepsilon > 0$.*

We now present a generalization of the facet guessing algorithm (Algorithm 1). The facet guessing algorithm [7] succeeds because the second best vertex is the optimal vertex on some facet of the polytope. Thus by solving an LP for every facet of the polytope, we guarantee that the second best solution is the optimal solution to one of the LPs. Outputting the top two vertices found will be the 2-best vertices of the LP. By a similar argument, if we enumerate lower dimensional faces of the polytope (which can be done by changing inequality constraints to equality constraints), we can find the M-best vertices by solving $O(m^{M-1})$ linear programs, where m is the number of constraints of the linear program. A similar idea was used implicitly in [1].

Algorithm 1 Generalized Facet Guessing

Input: an LP $\{\min c^T x : Ax \leq b\}$ with $A \in \mathbb{R}^{n \times m}$; a positive integer M
for all $I \subseteq \{1, 2, \ldots, m\}$ of size $M - 1$:
    $v_I \leftarrow \min c^T x : Ax \leq b, A_I x = b_I$
return the M best unique values from the v_I's

The hardness results do not prohibit finding the M best vertices for $M = O(\log n)$, so it still may be possible that we can find a super-constant number of vertices in polynomial time.

The advantage of Algorithm 1 over the extension complexity result of [1] is that it outputs many candidate suggestions for optimal vertices, suggesting a natural heuristic for finding more than a constant number of solutions. We may not have to enumerate all $O(m^{M-1})$ faces in order to return the M-best vertices found.

In [1], they show that if the polytope has all vertices contained in $\{0, 1\}^n$, we can find the M best for $M = \text{poly}(n)$ by constructing a series of polytopes with polynomial extension complexity that have the currently found solutions removed.

An alternative algorithm is the following: (1) we can find the second best vertex using the facet guessing algorithm and (2) we can separate the vertices into two sets such that one contains the optimal solution and the other contains the second best solution. The separation is by creating two new polytopes, one with $x_i = 0$ and one with $x_i = 1$, where i is chosen to separate the optimal and second best solution. This allows us to construct an arbitrary M best solver for any M by using techniques in [13].

However, these approaches will fail if there are fractional vertices amongst the top integral vertices, which is often the case in the LP relaxation of MAP in graphical models. In the following section, we discuss how to skip over these vertices.

4 Finding the M-best integral vertices

Another application of the M-best LP problem is to search for a vertex with a certain property. For example, the MAP problem in graphical models is often solved by an LP relaxation that is tightened until an integral vertex is found. It is a major open problem to understand why this approach is successful in practice compared to the theoretical worst case [19]. Success conditions do exist for certain special cases; see [16, 22, 23] for an incomplete list.

Assuming that the optimal integral vertex is among the $\text{poly}(n)$-best vertices and the polytope is contained in the unit cube, then we can find it in polynomial time with the following coordinate branching algorithm, which utilizes the classic branching technique often used when solving ILPs.

Algorithm 2 Coordinate Branching Algorithm

Input: an LP $\{\min c^T x : Ax \leq b, 0 \leq x \leq 1\}$; a positive integer M
represent branch as (v, I_0, I_1), meaning v is optimal for the LP with $x_{I_0} = 0$ and $x_{I_1} = 1$.
$v \leftarrow$ optimal vertex to original LP
$B \leftarrow \{(v_1, \emptyset, \emptyset)\}$
while we have not found optimal integer vertex:
    $(v, I_0, I_1) \leftarrow$ branch corresponding to the best vertex in B
    if v is integral:
        return v
    else:
        find a fractional coordinate v_i
        replace (v, I_0, I_1) in B with $(v_0, I_0 \cup \{i\}, I_1)$ and $(v_1, I_0, I_1 \cup \{i\})$

This generalizes the result in [7], which was able to solve the same problem in the specific case of LDPC decoding.

The algorithm starts out with with the optimal solution, then branches on a fractional coordinate. It then finds the maximum solution over all the current branches and again branches on a fractional coordinate, repeating this process until the optimal integral solution is found. It can also be seen as a facet guessing algorithm, as it guesses facets based on the coordinates of the current fractional solution.

Its success follows from the following invariants: (1) a vertex is never in more than one branch, (2) every integral vertex is in exactly one branch, (3) no new vertices are ever introduced, and (4) after every iteration we eliminate at least one fractional vertex with a lower cost than the optimal integral vertex. Thus we have the following theorem.

Theorem 2. *Given a bounded linear program of the form $\{\min c^T x : Ax \leq b, 0 \leq x \leq 1\}$, if the optimal integral vertex is among the M-best vertices, then the Coordinate Branching Algorithm (Algorithm 2) will terminate with the optimal integral vertex in time $O(MT_{LP})$, where T_{LP} is the time to optimize a linear program.*

Further, by combining the Coordinate Branching Algorithm with the algorithm we describe for $\{0, 1\}$-polytopes in the previous section, we can find the M-best integral solutions under the assumption that all M are contained amongst the $\mathrm{poly}(n)$-best vertices. This leads to the following corollary.

Corollary 3. *Given a bounded linear program of the form $\{\min c^T x : Ax \leq b, 0 \leq x \leq 1\}$, if the M optimal integral vertices is among the $\mathrm{poly}(n)$-best vertices, then we can find these M integral vertices in polynomial time.*

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 1018829, 1344179, 1344364, 1407278, 1422549 and ARO YIP W911NF-14-1-0258.

References

[1] G. Angulo, S. Ahmed, S. S. Dey, and V. Kaibel. Forbidden vertices. *Mathematics of Operations Research*, 2014.

[2] S. Arora, C. Daskalakis, and D. Steurer. Message-passing algorithms and improved LP decoding. *IEEE Transactions on Information Theory*, 2012.

[3] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-best solutions in Markov random fields. In *ECCV*, 2012.

[4] D. Burshtein. Iterative approximate linear programming decoding of ldpc codes with linear complexity. *IEEE Transactions on Information Theory*, 2009.

[5] M. Chertkov and M. G. Stepanov. An efficient pseudocodeword search algorithm for linear programming decoding of ldpc codes. *IEEE Transactions on Information Theory*, 2008.

[6] C. Daskalakis, A. G. Dimakis, R. M. Karp, and M. J. Wainwright. Probabilistic analysis of linear programming decoding. *IEEE Transactions on Information Theory*, 2008.

[7] A. G. Dimakis, A. A. Gohari, and M. J. Wainwright. Guessing facets: Polytope structure and improved LP decoder. *IEEE Transactions on Information Theory*, 2009.

[8] A. G. Dimakis, J. Wang, and K. Ramchandran. Unequal growth codes: Intermediate performance and unequal error protection for video streaming. In *IEEE Workshop on Multimedia Signal Processing*, 2007.

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

[10] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright. Lp decoding corrects a constant fraction of errors. *IEEE Transactions on Information Theory*, 2007.

[11] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 2005.

[12] N. Flerova, R. Marinescu, and R. Dechter. Searching for the m best solutions in graphical models. *Journal of Artificial Intelligence Research*, 2016.

[13] M. Fromer and A. Globerson. An LP view of the M-best MAP problem. In *NIPS*, 2009.

[14] K. Fukuda, T. M. Liebling, and F. Margot. Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Computational Geometry*, 1997.

[15] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *NIPS*, 2012.

[16] T. Jebara. Map estimation, message passing, and perfect graphs. In *UAI*, 2009.

[17] R. Koetter, W.-C. W. Li, P. O. Vontobel, and J. L. Walker. Characterizations of pseudo-codewords of (low-density) parity-check codes. *Advances in Mathematics*, 2007.

[18] X. Liu, S. C. Draper, and B. Recht. Suppressing pseudocodewords by penalizing the objective of LP decoding. In *Information Theory Workshop*, 2012.

[19] O. Meshi, M. Mahdavi, A. Weller, and D. Sontag. Train and test tightness of LP relaxations in structured prediction. In *ICML*, 2016.

[20] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011.

[21] A. Pauls, D. Klein, and C. Quirk. Top-down k-best A* parsing. In *ACL*, 2010.

[22] M. J. Wainwright and M. I. Jordan. *Graphical models, exponential families, and variational inference*. Foundations and Trends in Machine Learning, 2008.

[23] A. Weller, M. Rowland, and D. Sontag. Tightness of LP relaxations for almost balanced models. In *AISTATS*, 2016.

[24] X. Zhang and P. H. Siegel. Adaptive cut generation algorithm for improved linear programming decoding of binary linear codes. *IEEE Transactions on Information Theory*, 2012.