Marginals-to-Models Reducibility

Tim Roughgarden
Stanford University
tim@cs.stanford.edu

Michael Kearns
University of Pennsylvania
mkearns@cis.upenn.edu

Abstract

We consider a number of classical and new computational problems regarding marginal distributions, and inference in models specifying a full joint distribution. We prove general and efficient reductions between a number of these problems, which demonstrate that algorithmic progress in inference automatically yields progress for "pure data" problems. Our main technique involves formulating the problems as linear programs, and proving that the dual separation oracle required by the ellipsoid method is provided by the target problem. This technique may be of independent interest in probabilistic inference.

1 Introduction

The movement between the specification of "local" marginals and models for complete joint distributions is ingrained in the language and methods of modern probabilistic inference. For instance, in Bayesian networks, we begin with a (perhaps partial) specification of local marginals or CPTs, which then allows us to construct a graphical model for the full joint distribution. In turn, this allows us to make inferences (perhaps conditioned on observed evidence) regarding marginals that were not part of the original specification.

In many applications, the specification of marginals is derived from some combination of (noisy) observed data and (imperfect) domain expertise. As such, even before the passage to models for the full joint distribution, there are a number of basic computational questions we might wish to ask of given marginals, such as whether they are consistent with *any* joint distribution, and if not, what the nearest consistent marginals are. These can be viewed as questions about the "data", as opposed to inferences made in models derived from the data.

In this paper, we prove a number of *general, polynomial time reductions* between such problems regarding data or marginals, and problems of inference in graphical models. By "general" we mean the reductions are not restricted to particular classes of graphs or algorithmic approaches, but show that any computational progress on the target problem immediately transfers to progress on the source problem. For example, one of our main results establishes that the problem of determining whether given marginals, whose induced graph (the "data graph") falls within some class $\mathcal{G}$, are consistent with any joint distribution reduces to the problem of MAP inference in Markov networks falling in the same class $\mathcal{G}$. Thus, for instance, we immediately obtain that the tractability of MAP inference in trees or tree-like graphs yields an efficient algorithm for marginal consistency in tree data graphs; and any future progress in MAP inference for other classes $\mathcal{G}$ will similarly transfer. Conversely, our reductions also can be used to establish negative results. For instance, for any class $\mathcal{G}$ for which we can prove the intractability of maginal consistency, we can immediately infer the intractability of MAP inference as well.

There are a number of reasons to be interested in such problems regarding marginals. One, as we have already suggested, is the fact that given marginals may not be consistent with any joint

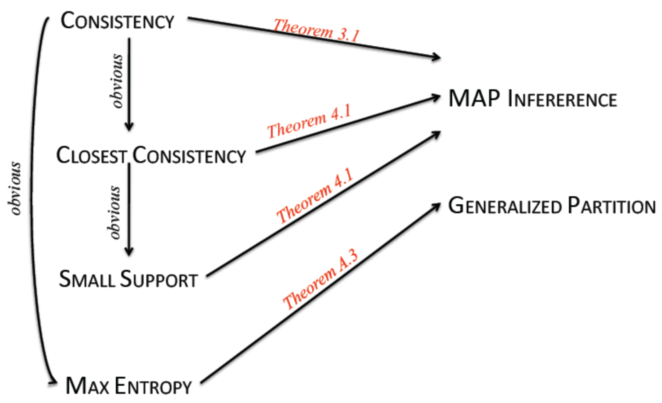

Figure 1: Summary of main results. Arrows indicate that the source problem can be reduced to the target problem for any class of graphs $\mathcal{G}$, and in polynomial time. Our main results are the left-to-right arrows from marginals-based problems to Markov net inference problems.

distribution, due to noisy observations or faulty domain intuitions, [1] and we may wish to know this before simply passing to a joint model that forces or assumes consistency. At the other extreme, given marginals may be consistent with *many* joint distributions, with potentially very different properties. [2] Rather than simply selecting one of these consistent distributions in which to perform inference (as would typically happen in the construction of a Markov or Bayes net), we may wish to reason over the entire class of consistent distributions, or optimize over it (for instance, choosing to maximize or minimize independence).

We thus consider four natural algorithmic problems involving (partially) specified marginals:

- CONSISTENCY: Is there any joint distribution consistent with given marginals?
- CLOSEST CONSISTENCY: What are the consistent marginals closest to given inconsistent marginals?
- MAX ENTROPY: What is the maximum entropy distribution closest to given marginals?
- SMALL SUPPORT Is there a small-support distribution close to given marginals?

The consistency problem has been studied before as the membership problem for the marginal polytope (see Related Work); in the case of inconsistency, the closest consistency problem seeks the minimal perturbation to the data necessary to recover coherence. While maximum entropy is a staple of probabilistic inference, the small support problem is also of basic interest: it can be informally viewed as attempting to minimize independence or randomization, and thus is a natural contrast to maximum entropy. (It is also worth noting that small support distributions arise naturally through the joint behavior of no-regret algorithms in game-theoretic settings [1].) More generally, when there are many consistent distributions, which one should be singled out? There are, of course, well-known reasons to prefer the maximum entropy distribution. But this is not the only approach. For example, consider the three features “votes Republican”, “supports universal healthcare”, and “supports tougher gun control”, and suppose the single marginals are 0.5, 0.5, 0.5. The maximum entropy distribution is uniform over the 8 possibilities. We might expect reality to hew closer to a small support distribution, perhaps even 50/50 over the two vectors 100 and 011.

We also consider two standard algorithmic inference problems on full joint distributions (models):

---

[1] As a simple example, consider just three random variables connected in a triangle, in which each pairwise marginal specifies that the settings (0,1) and (1,0) each occurs with probability $1/2$. Since each variable setting must disagree with each of its neighbors, there is no consistent joint distribution.

[2] For example, consider random variables X, Y, Z in which the data or marginal graph is a line connecting X and Y, and Y and Z, and in which each of these pairwise marginals specifies that all four binary settings are equally likely. One consistent distribution flips a fair coin independently for each variable; but another flips one coin for Y, a second for X, but then sets $Z = X$. The former maximizes entropy while the latter minimizes support size.

- MAP INFERENCE: What is the MAP joint assignment in a given Markov network?
- GENERALIZED PARTITION: What is the normalizing constant of a given Markov network, possibly after conditioning on the value of one vertex or edge?

Each of these problems is parameterized by a class of graphs $\mathcal{G}$ — for the marginals problems, this is the graph induced by the pairwise marginals, while for the models problems, it is the graph of the given Markov network. All of our reductions are of the form "for any class $\mathcal{G}$, if there is a polynomial time algorithm for solving inference problem B for (model) graphs in $\mathcal{G}$, there is a polynomial time algorithm for marginals problem A for (marginal) graphs in $\mathcal{G}$" — that is, A reduces to B. Our main results, which are summarized in Figure 1, can be stated informally as follows:

- CONSISTENCY reduces to MAP INFERENCE.
- CLOSEST CONSISTENCY reduces to MAP INFERENCE.
- SMALL SUPPORT reduces to MAP INFERENCE.
- MAX ENTROPY reduces to GENERALIZED PARTITION.

While connections between some of these problems are known for *specific classes* of graphs — most notably in trees, where all of these problems are tractable and rely on common underlying algorithmic approaches such as dynamic programming — the novelty of our results relies in their generality, showing the above reductions hold for *any* class of graphs.

All of our reductions share a common and powerful technique: the use of the ellipsoid method for Linear Programming (LP), with the key step being the articulation of an appropriate separation oracle. The first three problems we consider have a straightforward LP formulation which will typically have a number of variables that is equal to the number of joint settings, and therefore exponential in the number of variables; for the MAX ENTROPY problem, there is an analogous convex program formulation. Since our goal is to run in time polynomial in the input length (the number and size of given marginals), the straightforward LP formulation will not suffice. However, by passing to the dual LP, we instead obtain an LP with only a polynomial number of variables, but an exponential number of constraints that can be represented implicitly. For each of the reductions above, we show that the required separation oracle for these implicit constraints is provided exactly by the corresponding inference problem (MAP INFERENCE or GENERALIZED PARTITION). We believe this technique may be of independent interest and have other applications in probabilistic inference.

It is perhaps surprising that in the study of problems strictly addressing properties of given marginals (which have received relatively little attention in the graphical models literature historically), problems of inference in full joint models (which have received great attention) should arise so naturally and generally. For the marginal problems, our reductions (via the ellipsoid method) effectively create a series of "fictitious" Markov networks such that the solutions to corresponding inference problems (MAP INFERENCE and GENERALIZED PARTITION) indirectly lead to a solution to the original marginal problems.

Related Work: As we have noted, the literature on graphical models and probabilistic inference is rife with connections between some of the problems we study here for specific classes of graphical models (such as trees or otherwise sparse structures), and under specific algorithmic approaches (such as dynamic programming or message-passing algorithms more generally, and various forms of variational inference); see [2, 3, 4] for good overviews. In contrast, here we develop general and efficient reductions between marginal and inference problems that hold regardless of the graph structure or algorithmic approach; we are not aware of prior efforts in this vein. Some of the problems we consider are also either new or have been studied very little, such as closest consistency and small support distributions.

Perhaps most closely related in spirit to our interests are [5] and [6, 7]. These works all provide reductions of some form, but not ones that are both general (independent of graph structure) and polynomial time. However, they do suggest both the possibility and interest in such stronger reductions. The paper [5] discusses and provides heuristic reductions between MAP INFERENCE and GENERALIZED PARTITION. The work in [6, 7] makes the technical point that maximizing entropy subject to an (approximate) consistency condition yields a distribution that can be represented as a Markov network over the graph induced by the original data or marginals, but does not provide a formal complexity-theoretic treatment nor worst-case polynomial time bounds. More generally, as far as we are aware, there has been essentially no formal complexity analysis (i.e., worst-case polynomial-time guarantees) for algorithms that compute max-entropy distributions. Such guarantees do not follow automatically from convexity, for two reasons. The first, which has been addressed in previous work by the community, is the exponential number of decision variables (circumvented via a suitable separation oracle). The second, which does not seem to have been previously addressed, is the need to bound the diameter of the search space (e.g., if the optimal Lagrange variables are too big, it will take too long to find them). Bounding the search space requires using special properties of the max entropy problem, and our results elucidate the necessary argument.

The consistency problem has been studied before as the membership problem for the marginal polytope. For example, [8] shows that finding the MAP assignment for Markov random fields with pairwise potentials can be cast as an integer linear program over the marginal polytope — that is, algorithms for the consistency problem are useful subroutines for inference. As far as we are aware, our work is the first to show a converse, that inference algorithms are useful subroutines for decision and optimization problems for the marginal polytope. Furthermore, previous polynomial-time solutions to consistency generally give a compact (polynomial-size) description of the marginal polytope. Our approach dodges this ambitious requirement, in that it only needs a polynomial-time separation oracle (which, for this problem, turns out to be MAP inference). There are many combinatorial optimization problems with no compact LP formulation that admit polynomial-time ellipsoid-based algorithms (like nonbipartite matching, with its exponentially many odd cycle inequalities).

2 Preliminaries

2.1 Problem Definitions

For clarity of exposition, we focus on the pairwise case in which every marginal involves at most two variables. [3] Denote the underlying random variables by $X_1, \ldots, X_n$, which we assume have range $[k] = \{0, 1, 2, \ldots, k\}$. The input is at most one real-valued *single marginal* μ_{is} for every variable $i \in [n]$ and value $s \in [k]$, and at most one real-valued *pairwise marginal* μ_{ijst} for every ordered variable pair $i, j \in [n] \times [n]$ with $i < j$ and every pair $s, t \in [k]$. The *data graph* induced by a set of marginals has one vertex per random variable X_i, and an undirected edge (i, j) if and only if at least one of the given pairwise marginals involves the variables X_i and X_j. Let M_1 and M_2 denote the indices (i, s) and (i, j, s, t) of the given single and pairwise marginals, and $m = |M_1| + |M_2|$ the total number of marginals. Let $A = [k]^n$ denote the space of all possible variable assignments. We say that the given marginals μ are *consistent* if there exists a (joint) probability distribution consistent with all of them (i.e., that induces the marginals μ).

With these basic definitions, we can now give formal definitions for the marginals problems we consider. Let $\mathcal{G}$ denote an arbitrary class of undirected graphs.

- CONSISTENCY ($\mathcal{G}$): Given marginals μ such that the induced data graph falls in $\mathcal{G}$, are they consistent?
- CLOSEST CONSISTENCY ($\mathcal{G}$): Given (possibly inconsistent) marginals μ on $M_1 \cup M_2$ such that the induced data graph falls in $\mathcal{G}$, compute the consistent marginals ν minimizing $||\nu - \mu||_1$.
- SMALL SUPPORT ($\mathcal{G}$): Given (consistent or inconsistent) marginals μ such that the induced data graph falls in $\mathcal{G}$, compute a distribution that has a polynomial-size supports and marginals ν that minimize $||\nu - \mu||_1$.
- MAX ENTROPY ($\mathcal{G}$): Given (consistent or inconsistent) marginals μ such that the induced data graph falls in $\mathcal{G}$, compute the maximum entropy distribution that has marginals ν that minimize $||\nu - \mu||_1$.

---

[3] For ease of exposition, throughout we shall consider the case in which only singleton and pairwise marginals are given. All of our results generalize to the case of higher-order marginals in a straightforward manner, and also to the setting in which marginals may be only partially specified.

It is important to emphasize that all of the problems above are "model-free", in that we do not assume that the marginals are consistent with, or generated by, any particular model (such as a Markov network). They are simply given marginals, or "data".

For each of these problems, our interest is in algorithms whose running time is polynomial in the size of the input μ. The prospects for this depend strongly on the class $\mathcal{G}$, with tractability generally following for "nice" classes such as tree or tree-like graphs, and intractability resulting in the most general cases. Our contribution is in showing a strong connection between tractability for these marginals problems and the following inference problems for *any* class G.

- MAP INFERENCE ($\mathcal{G}$): Given a Markov network whose graph falls in $\mathcal{G}$, find the maximum a posteriori (MAP) or most probable joint assignment.[4]
- GENERALIZED PARTITION: Given a Markov network whose graph falls in $\mathcal{G}$, compute the partition function or normalization constant for the full joint distribution, possibly after conditioning on the value of a single vertex or edge.[5]

2.2 The Ellipsoid Method for Linear Programming

Our algorithms for the CONSISTENCY, CLOSEST CONSISTENCY, and SMALL SUPPORT problems use linear programming. There are a number of algorithms that solve explictly described linear programs in time polynomial in the description size. Our problems, however, pose an additional challenge: the obvious linear programming formulation has size exponential in the parameters of interest. To address this challenge, we turn to the ellipsoid method [9], which can solve in polynomial time linear programs that have an exponential number of implicitly described constraints, provided there is a polynomial-time "separation oracle" for these constraints. The ellipsoid method is discussed exhaustively in [10, 11]; we record in this section the facts necessary for our results.

Definition 2.1 (Separation Oracle) *Let* $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_1^T\mathbf{x} \le b_1, \dots, \mathbf{a}_m^T\mathbf{x} \le b_m\}$ *denote the feasible region of* m *linear constraints in* n *dimensions. A* separation oracle *for* $\mathcal{P}$ *is an algorithm that takes as input a vector* $\mathbf{x} \in \mathbb{R}^n$*, and either (i) verifies that* $\mathbf{x} \in \mathcal{P}$*; or (ii) returns a constraint* i *such that* $\mathbf{a}_i^t\mathbf{x} > b_i$*. A* polynomial-time *separation oracle runs in time polynomial in* n*, the maximum description length of a single constraint, and the description length of the input* $\mathbf{x}$.

One obvious separation oracle is to simply check, given a candidate solution $\mathbf{x}$, each of the m constraints in turn. More interesting and relevant are constraint sets that have size exponential in the dimension n but admit a polynomial-time separation oracle.

Theorem 2.2 (Convergence Guarantee of the Ellipsoid Method [9]) *Suppose the set* $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_1^T\mathbf{x} \le b_1, \dots, \mathbf{a}_m^T\mathbf{x} \le b_m\}$ *admits a polynomial-time separation oracle and* $\mathbf{c}^T\mathbf{x}$ *is a linear objective function. Then, the ellipsoid method solves the optimization problem* $\{\max \mathbf{c}^T\mathbf{x} : \mathbf{x} \in \mathcal{P}\}$ *in time polynomial in* n *and the maximum description length of a single constraint or objective function. The method correctly detects if* $\mathcal{P} = \emptyset$*. Moreover, if* $\mathcal{P}$ *is non-empty and bounded, the ellipsoid method returns a vertex of* $\mathcal{P}$*.*[6]

Theorem 2.2 provides a general reduction from a problem to an intuitively easier one: *if* the problem of verifying membership in $\mathcal{P}$ can be solved in polynomial time, *then* the problem of optimizing an arbitrary linear function over $\mathcal{P}$ can also be solved in polynomial time. This reduction is "many-to-one," meaning that the ellipsoid method invokes the separation oracle for $\mathcal{P}$ a large (but polynomial) number of times, each with a different candidate point $\mathbf{x}$. See Appendix A.1 for a high-level description of the ellipsoid method and [10, 11] for a detailed treatment.

The ellipsoid method also applies to convex programming problems under some additional technical conditions. This is discussed in Appendix A.2 and applied to the MAX ENTROPY problem in Appendix A.3.

---

[4] Formally, the input is a graph $G = (V, E)$ with a potential $\phi_i(s)$ and $\phi_{ij}(s,t)$ for each vertex $i \in V$ and edge $(i,j) \in E$, and each value $s \in [k] = \{0, 1, 2 \dots, k\}$ and pair $s, t \in [k] \times [k]$ of values. The MAP assignment maximizes $P(a) := \prod_{i \in V} \phi_i(a_i) \prod_{(i,j) \in E} \phi_{ij}(a_i, a_j)$ over all assignments $a \in [k]^V$.

[5] Formally, given a Markov network, compute $\sum_{a \in [k]^n} P(a)$; $\sum_{a\,:\,a_i = s} P(a)$ for a given i, s; or $\sum_{a\,:\,a_i = s, a_j = t} P(a)$ for a given i, j, s, t.

[6] A *vertex* is a point of $\mathcal{P}$ that satisfies with equality n linearly independent constraints.

3 CONSISTENCY Reduces to MAP INFERENCE

The goal of this section is to reduce the CONSISTENCY problem for data graphs in the family $\mathcal{G}$ to the MAP INFERENCE problem for networks in $\mathcal{G}$.

Theorem 3.1 (Main Result 1) *Let $\mathcal{G}$ be a set of graphs. If the the* MAP INFERENCE ($\mathcal{G}$) *problem can be solved in polynomial time, then the* CONSISTENCY ($\mathcal{G}$) *problem can be solved in polynomial time.*

We begin with a straightforward linear programming formulation of the CONSISTENCY problem.

Lemma 3.2 (Linear Programming Formulation) *An instance of the* CONSISTENCY *problem admits a consistent distribution if and only if the following linear program (P) has a solution:*

$$
\begin{array}{lll}
(P) & \max\limits_{\mathbf{p}} & 0 \\
\textit{subject to:} & & \\
& \sum_{a\in A: a_i=s} p_a = \mu_{is} & \textit{for all } (i,s)\in M_1 \\
& \sum_{a\in A: a_i=s, a_j=t} p_a = \mu_{ijst} & \textit{for all } (i,j,s,t)\in M_2 \\
& \sum_{a\in A} p_a = 1 & \\
& p_a \geq 0 & \textit{for all } a\in A.
\end{array}
$$

Solving (P) using the ellipsoid method (Theorem 2.2), or any other linear programming method, requires time at least $|A| = (k+1)^n$, the number of decision variables. This is generally exponential in the size of the input, which is proportional to the number m of given marginals.

A ray of hope is provided by the fact that the number of *constraints* of the linear program in Lemma 3.2 is equal to the number of marginals. With an eye toward applying the ellipsoid method (Theorem 2.2), we consider the dual linear program. We use the following notation. Given a vector $\mathbf{y}$ indexed by $M_1 \cup M_2$, we define

$$\mathbf{y}(a) = \sum_{(i,s)\in M_1\,:\,a_i=s} y_{is} + \sum_{(i,j,s,t)\in M_2\,:\,a_i=s,a_j=t} y_{ijst} \tag{1}$$

for each assignment $a \in A$, and

$$\mu^T\mathbf{y} = \sum_{(i,s)\in M_1} \mu_{is}y_{is} + \sum_{(i,j,s,t)\in M_2} \mu_{ijst}y_{ijst}. \tag{2}$$

Strong linear programming duality implies the following.

Lemma 3.3 (Dual Linear Programming Formulation) *An instance of the* CONSISTENCY *problem admits a consistent distribution if and only if the optimal value of the following linear program (D) is 0:*

$$
\begin{array}{lll}
(D) & \max\limits_{\mathbf{y},z} & \mu^T\mathbf{y} + z \\
\textit{subject to:} & & \\
& \mathbf{y}(a) + z \leq 0 & \textit{for all } a\in A \\
& \mathbf{y}, z \ \textit{unrestricted.} &
\end{array}
$$

The number of variables in (D) — one per constraint of the primal linear program — is polynomial in the size of the CONSISTENCY input.

What use is the MAP INFERENCE problem for solving the CONSISTENCY problem? The next lemma forges the connection.

Lemma 3.4 (Map Inference as a Separation Oracle) *Let $\mathcal{G}$ be a set of graphs and suppose that the* MAP INFERENCE ($\mathcal{G}$) *problem can be solved in polynomial time. Consider an instance of the* CONSISTENCY *problem with a data graph in $\mathcal{G}$, and a candidate solution $\mathbf{y}, z$ to the corresponding* *linear program (D). Then, there is a polynomial-time algorithm that checks whether or not there is an assignment $a \in A$ that satisfies*

$$\sum_{(i,s)\in M_1\,:\,a_i=s} y_{is} + \sum_{(i,j,s,t)\in M_2\,:\,a_i=s,a_j=y} y_{ijst} > -z, \tag{3}$$

and produces such an assignment if one exists.

Proof: The key idea is to interpret $\mathbf{y}$ as the log-potentials of a Markov network. Precisely, construct a Markov network N as follows. The vertex set V and edge set E correspond to the random variables and edge set of the data graph of the CONSISTENCY instance. The potential function at a vertex i is defined as $\phi_i(s) = \exp\{y_{is}\}$ for each value $s \in [k]$. The potential function at an edge (i,j) is defined as $\phi_{ij}(s,t) = \exp\{y_{ijst}\}$ for $(s,t) \in [k] \times [k]$. For a missing pair $(i,s) \notin M_1$ or 4-tuple $(i,j,s,t) \notin M_2$, we define the corresponding potential value $\phi_i(s)$ or $\phi_{ij}(st)$ to be 1. The underlying graph of N is the same as the data graph of the given CONSISTENCY instance and hence is a member of $\mathcal{G}$.

In the distribution induced by N, the probability of an assignment $a \in [k]^n$ is, by definition, proportional to

$$\left(\prod_{i\in V\,:\,(i,a_i)\in M_1} \exp\{y_{ia_i}\}\right)\left(\prod_{(i,j)\in E\,:\,(i,j,a_i,a_j)\in M_2} \exp\{y_{ija_ia_j}\}\right) = \exp\{\mathbf{y}(a)\}.$$

That is, the MAP assignment for the Markov network N is the assignment that maximizes the left-hand size of (3).

Checking if some assignment $a \in A$ satisfies (3) can thus be implemented as follows: compute the MAP assignment a^* for N — by assumption, and since the graph of N lies in $\mathcal{G}$, this can be done in polynomial time; return a^* if it satisfies (3), and otherwise conclude that no assignment $a \in A$ satisfies (3). ■

All of the ingredients for the proof of Theorem 3.1 are now in place.

Proof of Theorem 3.1: Assume that there is a polynomial-time algorithm for the MAP INFERENCE ($\mathcal{G}$) problem with the family $\mathcal{G}$ of graphs, and consider an instance of the CONSISTENCY problem with data graph $G \in \mathcal{G}$. Deciding whether or not this instance has a consistent distribution is equivalent to solving the program (D) in Lemma 3.3. By Theorem 2.2, the ellipsoid method can be used to solve (D) in polynomial time, provided the constraint set admits a polynomial-time separation oracle. Lemma 3.4 shows that the relevant separation oracle is equivalent to computing the MAP assignment of a Markov network with graph $G \in \mathcal{G}$. By assumption, the latter problem can be solved in polynomial time. ■

We defined the CONSISTENCY problem as a decision problem, where the answer is "yes" or no." For instances that admit a consistent distribution, we can also ask for a succinct representation of a distribution that witnesses the marginals' consistency. We next strengthen Theorem 3.1 by showing that for consistent instances, under the same hypothesis, we can compute a small-support consistent distribution in polynomial time. See Figure 2 for the high-level description of the algorithm.

Theorem 3.5 (Small-Support Witnesses) *Let $\mathcal{G}$ be a set of graphs. If the the* MAP INFERENCE *($\mathcal{G}$) problem can be solved in polynomial time, then for every consistent instance of the* CONSISTENCY *($\mathcal{G}$) problem with $m = |M_1| + |M_2|$ marginals, a consistent distribution with support size at most $m+1$ can be computed in polynomial time.*

Proof: Consider a consistent instance of CONSISTENCY with data graph $G \in \mathcal{G}$. The algorithm of Theorem 3.1 concludes by solving the dual linear program of Lemma 3.3 using the ellipsoid method. This method runs for a polynomial number K of iterations, and each iteration generates one new inequality. At termination, the algorithm has identified a "reduced dual linear program", in which a set of only K out of the original $(k+1)^n$ constraints is sufficient to prove the optimality of its solution. By strong duality, the corresponding "reduced primal linear program," obtained from the linear program in Lemma 3.2 by retaining only the decision variables corresponding to the K

---

1. Solve the dual linear program (D) (Lemma 3.3) using the ellispoid method (Theorem 2.2), using the given polynomial-time algorithm for MAP INFERENCE ($\mathcal{G}$) to implement the ellipsoid separation oracle (see Lemma 3.4).
2. If the dual (D) has a nonzero (and hence, unbounded) optimal objective function value, then report "no consistent distributions" and halt.
3. Explicitly form the reduced primal linear program (P-red), obtained from (P) by retaining only the variables that correspond to the dual inequalities generated by the separation oracle in Step 1.
4. Solve (P-red) using a polynomial-time linear programming algorithm that returns a vertex solution, and return the result.

Figure 2: High-level description of the polynomial-time reduction from CONSISTENCY ($\mathcal{G}$) to MAP INFERENCE ($\mathcal{G}$) (Steps 1 and 2) and postprocessing to extract a small-support distribution that witnesses consistent marginals (Steps 3 and 4).

---

reduced dual constraints, has optimal objective function value 0. In particular, this reduced primal linear program is feasible.

The reduced primal linear program has a polynomial number of variables and constraints, so it can be solved by the ellipsoid method (or any other polynomial-time method) to obtain a feasible point $\mathbf{p}$. The point $\mathbf{p}$ is an explicit description of a consistent distribution with support size at most K. To improve the support size upper bound from K to $m + 1$, recall from Theorem 2.2 that $\mathbf{p}$ is a vertex of the feasible region, meaning it satisfies K linearly independent constraints of the reduced primal linear program with equality. This linear program has at most one constraint for each of the m given marginals, at most one normalization constraint $\sum_{a \in A} p_a = 1$, and nonnegativity constraints. Thus, at least $K - m - 1$ of the constraints that $\mathbf{p}$ satisfies with equality are nonnegativity constraints. Equivalently, it has at most $m + 1$ strictly positive entries. ■

4 CLOSEST CONSISTENCY, SMALL SUPPORT Reduce to MAP INFERENCE

This section considers the CLOSEST CONSISTENCY and SMALL SUPPORT problems. The input to these problem is the same as in the CONSISTENCY problem — single marginals μ_{is} for $(i, s) \in M_1$ and pairwise marginals μ_{ijst} for $(i, j, s, t) \in M_2$. The goal is to compute sets of marginals $\{\nu_{is}\}_{M_1}$ and $\{\nu_{ijst}\}_{M_2}$ that are consistent and, subject to this constraint, minimize the ℓ_1 norm $||\mu - \nu||_1$ with respect to the given marginals. An algorithm for the CLOSEST CONSISTENCY problem solves the CONSISTENCY problem as a special case, since a given set of marginals is consistent if and only if the corresponding CLOSEST CONSISTENCY problem has optimal objective function value 0. Despite this greater generality, the CLOSEST CONSISTENCY problem also reduces in polynomial time to the MAP INFERENCE problem, as does the still more general SMALL SUPPORT problem.

Theorem 4.1 (Main Result 2) *Let $\mathcal{G}$ be a set of graphs. If the the* MAP INFERENCE ($\mathcal{G}$) *problem can be solved in polynomial time, then the* CLOSEST CONSISTENCY ($\mathcal{G}$) *problem can be solved in polynomial time. Moreover, a distribution consistent with the optimal marginals with support size at most* $3m + 1$ *can be computed in polynomial time, where* $m = |M_1| + |M_2|$ *denotes the number of marginals.*

The formulation of the CLOSEST CONSISTENCY ($\mathcal{G}$) problem has linear constraints — the same as those in Lemma 3.2, except with the given marginals μ replaced by the computed consistent marginals ν — but a nonlinear objective function $||\mu - \nu||_1$. We can simulate the absolute value functions in the objective by adding a small number of variables and constraints. We provide details and the proof of Theorem 4.1 in Appendix A.4.

References

[1] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[2] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[3] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1), 2008.

[4] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[5] T. Hazan and T. Jaakkola. On the partition function and random maximum a-posteriori perturbations. *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[6] J. K. Johnson, V. Chandrasekaran, and A. S. Willsky. Learning markov structure by maximum entropy relaxation. In *11th International Conference in Artificial Intelligence and Statistics (AISTATS 2007)*, 2007.

[7] V. Chandrasekaran, J. K. Johnson, and A. S. Willsky. Maximum entropy relaxation for graphical model selection given inconsistent statistics. In *IEEE Statistical Signal Processing Workshop (SSP 2007)*, 2007.

[8] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *Neural Information Processing Systems (NIPS)*, 2007.

[9] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[10] A. Ben-Tal and A. Nemirovski. Optimization iii. Lecture notes, 2012.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988. Second Edition, 1993.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge, 2004.

[13] M. Singh and N. Vishnoi. Entropy, optimization and counting. *arXiv*, (1304.8108), 2013.

A Supplementary Material

A.1 High-Level Description of the Ellipsoid Method

Very roughly, the method works as follows. First, the optimization problem is reduced to a feasibility problem by adding the constraint $\mathbf{c}^T\mathbf{x} \geq M$ to $\mathcal{P}$ and binary searching on M in an outer loop. For the feasibility problem, the method maintains an ellipsoid E that is an outer bound (i.e., superset of) $\mathcal{P}$. At every iteration i, the method uses the separation oracle to check if the centroid of the current ellipsoid E_i is feasible. If so, the method stops. If not, the separation oracle furnishes a violated constraint; intersecting this with E_i yields a " partial ellipsoid" E_i' that excludes the centroid of E_i. The next ellipsoid E_{i+1} is taken to be the minimum-volume one that encloses E_i'. A technical argument shows that the volume of E_{i+1} is significantly less than that of E_i, and this leads to the convergence bound. See [10, 11] for details.

A.2 The Ellipsoid Method for Convex Programming

As is well known, the MAX ENTROPY problem is a convex optimization problem. The ellipsoid method can be adapted to such problems under mild technical conditions. The following guarantee, which can be derived from more general results (e.g. [10, 11]), is sufficient for our purposes. It states that, provided the relevant magnitudes of feasible solutions and objective function values are at most exponential, then given polynomial-time oracles for evaluating the objective function of a convex program and its gradient, the ellipsoid method can solve the program in time polynomial in the input size and the desired precision.

Theorem A.1 (The Ellipsoid Method for Convex Programs) *Consider an unconstrained minimization problem of the form* $\{\inf_{\mathbf{y}} f(\mathbf{y}) \,:\, \mathbf{y} \in \mathbb{R}^n\}$ *and suppose that:*

1. *For an a priori known bound* R*, there is an optimal solution* $\mathbf{y}^*$ *with* $||\mathbf{y}||_\infty \leq R$.
2. *For every pair* $\mathbf{y}^1, \mathbf{y}^2$ *with* $||\mathbf{y}^1||_\infty$, $||\mathbf{y}^2||_\infty \leq R$, $|f(\mathbf{y}^1) - f(\mathbf{y}^2)| \leq K$.
3. *For every point* $\mathbf{y}$*, the objective function value* $f(\mathbf{y})$ *and its gradient* $\nabla f(\mathbf{y})$ *can be evaluated in time polynomial in* n *and description length of* $\mathbf{y}$.

Then, given $\epsilon > 0$*, the ellipsoid method computes a point* $\tilde{\mathbf{y}}$ *such that* $f(\tilde{\mathbf{y}}) < f(\mathbf{y}^*) + \epsilon$ *in time polynomial in* n, $\log R$, $\log K$*, and* $\log \frac{1}{\epsilon}$.

A.3 MAX ENTROPY Reduces to GENERALIZED PARTITION

This section describes a reduction from the MAX ENTROPY ($\mathcal{G}$) problem to the GENERALIZED PARTITION ($\mathcal{G}$) problem that is based on the ellipsoid method for convex programming (Theorem A.1). Before explaining the precise technical conditions that enable this reduction, we first ignore computational complexity issues and review some well-known theory about the MAX ENTROPY ($\mathcal{G}$) problem (see e.g. [12, 3] for more details).

The standard convex programming formulation of the MAX ENTROPY problem, subject to marginals $\{\mu\}$, is simply the linear program (P) of Lemma 3.2, augmented with the entropy objective function:

$$
\begin{array}{lll}
(P-ME) \quad \sup\limits_{\mathbf{p}} & \sum_{a \in A} p_a \ln \frac{1}{p_a} & \\
\text{subject to:} & & \\
& \sum_{a \in A : a_i = s} p_a = \mu_{is} & \text{for all } (i,s) \in M_1 \\
& \sum_{a \in A : a_i = s, a_j = t} p_a = \mu_{ijst} & \text{for all } (i,j,s,t) \in M_2 \\
& \sum_{a \in A} p_a = 1 & \\
& p_a \geq 0 & \text{for all } a \in A.
\end{array}
$$

Using the notation in (1) and (2), we can write the dual program to (P-ME) as:

$$(D - ME) \quad \inf_{\mathbf{y}} \quad \mu^T \mathbf{y} + \ln \textstyle\sum_{a \in A} \exp\{-\mathbf{y}(a)\}$$

$$\text{subject to:}$$

$$\mathbf{y} \ \text{unrestricted.}$$

Assuming there is a feasible solution to (P-ME) with full support (i.e., $p_a > 0$ for all $a \in A$, a form of the Slater condition), strong duality holds and both convex programs have identical optimal objective function values. Moreover, in this case the maximum entropy distribution can be exactly represented as a Markov network N with underlying graph equal to the data graph G of the marginals μ, with the negative of the log-potentials of N corresponding to the optimal dual solution, similar to the mapping in the proof of Lemma 3.4. (Missing marginals from μ are defined to have zero log-potential.)

Under what conditions can we implement this approach with an algorithm with running time time polynomial in the size of the MAX ENTROPY input? To see why convexity is not obviously enough, observe that the number of decision variables in the programs (P-ME) and (D-ME) is proportional to the number $(k+1)^n$ of variable assignments, with is typically exponential in the input size. We can, however, apply the ellipsoid method (Theorem A.1) to the dual program (D-ME) provided conditions 1.-3. are met by the problem. The next lemma connects the third condition in Theorem A.1 to the GENERALIZED PARTITION problem.

Lemma A.2 (Generalized Partition as a Gradient Oracle) *Let $\mathcal{G}$ be a set of graphs and suppose that the* GENERALIZED PARTITION *($\mathcal{G}$) problem can be solved in polynomial time. Consider an instance of the* MAX ENTROPY *problem with a data graph $G \in \mathcal{G}$, with corresponding dual convex program (D-ME). Then, there are algorithms for evaluating the objective function f of (D-ME) and its gradient ∇f that run in time polynomial in the size of the* MAX ENTROPY *instance and the magnitude of the evaluation point $\mathbf{y}$.*

Proof (sketch): Consider an evaluation point $\mathbf{y}$. Let N denote the Markov network for which $\mathbf{y}$ is the negative of the log-potentials, as above. The graph of N is identical to the data graph G of the given MAX ENTROPY instance. The term $\sum_a \exp\{-\mathbf{y}(a)\}$ in the objective function of (D-ME) is precisely the Partition function of N. Given this quantity, which by assumption can be computed in polynomial time, the rest of the objective function is straightforward to compute.

Second, a simple computation shows that the gradient component $\nabla f_{is}(\mathbf{y})$ at $\mathbf{y}$ corresponding to $(i, s) \in M_1$ is

$$\mu_{is} - \frac{\sum_{a \,:\, a_i = s} \mathbf{y}(a)}{\sum_{a \in A} \mathbf{y}(a)},$$

and similarly for components of the form μ_{ijst}. There are a polyomial number of components, and each is straightforward to compute given the assumed polynomial-time algorithm for the GENERALIZED PARTITION problem. ■

Singh and Vishnoi [13] studied entropy maximization over combinatorial structures subject to single marginals, and in their context identified an additional condition, essentially a quantitative strengthening of the Slater condition, that implies the first two conditions of Theorem A.1 for the dual program (D-ME). To adapt it to our settings, consider a set μ of marginals defined on $M_1 \cup M_2$. Let $\mathcal{P}(M_1, M_2)$ denote the set of all marginal vectors ν induced by probability distributions over A. For example, a set of marginals μ is consistent if and only if $\mu \in \mathcal{P}(M_1, M_2)$. More strongly, we say that μ is *η-strictly feasible* if the intersection of the ball with center μ and radius η with the set $\mathcal{P}(M_1, M_2)$ is contained in the relative interior of $\mathcal{P}(M_1, M_2)$. Our results below are interesting when η is at least some inverse polynomial function of the input size.

Following the proof in [13, Theorem 2.7] shows that, in every η-strictly feasible instance of MAX ENTROPY, there is an optimal dual solution $\mathbf{y}^*$ to (D-ME) with $||\mathbf{y}||_\infty \leq \frac{m}{\eta}$, where $m = |M_1| + |M_2|$ is the number of marginals. We can therefore take the constant R in Theorem A.1 to be $\frac{m}{\eta}$. Plugging this bound into the objective function of (D-ME) shows that we can take the constant K in Theorem A.1 to be exponential in $\frac{m}{\eta}$. Applying Theorem A.1 then gives the following reduction from the MAX ENTROPY problem to the GENERALIZED PARTITION problem.

Theorem A.3 (Main Result 3) *Let $\mathcal{G}$ be a set of graphs. If the* Generalized Partition ($\mathcal{G}$) *problem can be solved in polynomial time, then every η-strictly feasible instance of the* Max Entropy ($\mathcal{G}$) *problem can be solved up to error ϵ in time polynomial in the input size, $\frac{1}{\eta}$, and $\log \frac{1}{\epsilon}$.*

A.4 Proof of Theorem 4.1:
Closest Consistency and Small Support Reduce to MAP Inference

Lemma A.4 (LP Formulation for Closest Consistency**)** *The consistent marginals $\{\nu\}$ that minimize the ℓ_1 distance $||\nu - \mu||_1$ to the given marginals $\{\mu\}$ correspond to optimal solutions to the following mathematical program:*

$$(P - close) \quad \min_{\mathbf{p},\nu,\sigma} \quad \textstyle\sum_{(i,s)\in M_1} \sigma_{is} + \sum_{(i,j,s,t)\in M_2} \sigma_{ijst}$$

subject to:

$$\begin{array}{rl}
\sigma_{i,s} \geq \nu_{is} - \mu_{is} & \textit{for all } (i,s) \in M_1 \\
\sigma_{i,s} \geq \mu_{is} - \nu_{is} & \textit{for all } (i,s) \in M_1 \\
\sigma_{i,j,s,t} \geq \nu_{ijst} - \mu_{ijst} & \textit{for all } (i,j,s,t) \in M_2 \\
\sigma_{i,j,s,t} \geq \mu_{ijst} - \nu_{ijst} & \textit{for all } (i,j,s,t) \in M_2 \\
\sum_{a\in A: a_i = s} p_a = \nu_{is} & \textit{for all } (i,s) \in M_1 \\
\sum_{a\in A: a_i = s, a_j = t} p_a = \nu_{ijst} & \textit{for all } (i,j,s,t) \in M_2 \\
\sum_{a\in A} p_a = 1 & \\
p_a \geq 0 & \textit{for all } a \in A.
\end{array}$$

Proof: The constraints enforce the inequality $\sigma_{i,s} \geq |\nu_{i,s} - \mu_{i,s}|$ for all single marginals $(i,s) \in M_1$, and similarly for all pariwise marginals, at every feasible solution. The minimization objective ensures that equality holds for every such constraint at every optimal solution. Thus, optimal solutions to this linear program are in correpondence with those of the more straightforward nonlinear formulation. ■

We next need to pass to the linear programming dual to (P-close) to enable application of the ellipsoid method. The negative of this dual is, after some simplifications, as follows.

$$(D - close) \quad \max_{\mathbf{y},z} \quad \mu^T \mathbf{y} + z$$

subject to:

$$\begin{array}{rl}
\mathbf{y}(a) + z \leq 0 & \text{for all } a \in A \\
-1 \leq y_{is} \leq 1 & \text{for all } (i,s) \in M_1 \\
-1 \leq y_{ijst} \leq 1 & \text{for all } (i,j,s,t) \in M_2 \\
z \;\; \text{unrestricted.} &
\end{array}$$

Our proof of Theorem 4.1 now follows the same outline as that of Theorem 3.1.

Proof of Theorem 4.1: Assume that there is a polynomial-time algorithm for the MAP Inference ($\mathcal{G}$) problem with the family $\mathcal{G}$ of graphs, and consider an instance of the Closest Consistency with data graph in $G \in \mathcal{G}$. The ellipsoid method can be used to solve the dual linear program (D-close) in polynomial time, provided the constraint set admits a polynomial-time separation oracle. Given a candidate dual solution $\mathbf{y}, z$, the polynomially many constraints that enforce $|y| \leq 1$ can be checked explicitly, and the rest can be checked by computing the MAP assignment of a Markov network with graph G, as in Lemma 3.4. By assumption, this MAP inference problem can be solved in polynomial time.

To recover an optimal solution for the Closest Consistency instance, and to solve the Small Support problem, we proceed as in the proof of Theorem 3.5. We form a reduced version of (P-close), with variables corresponding to the (polynomially many) inequalities of (D-close) that were generated by the separation oracle. This reduced linear program has the same optimal objective function value as (P-close) and has polynomial size. The algorithm concludes by returning an optimal solution of this reduced linear program. Since this linear program has at most $3m + 1$ constraints other than the nonnegativity constraints, every optimal vertex solution has support size at most $3m + 1$. ■