Learnability of Parameter-Bounded Bayes Nets

Arnab Bhattacharyya [1] **Davin Choo** [1] **Sutanu Gayen** [2] **Dimitrios Myrisiotis** [3]

Abstract

Bayes nets are extensively used in practice to efficiently represent joint probability distributions over a set of random variables and capture dependency relations. In a seminal paper, Chickering et al. (JMLR 2004) showed that given a distribution $\mathbb{P}$, that is defined as the marginal distribution of a Bayes net, it is NP-hard to decide whether there is a parameter-bounded Bayes net that represents $\mathbb{P}$. They called this problem LEARN. In this work, we extend the NP-hardness result of LEARN and prove the NP-hardness of a promise search variant of LEARN, whereby the Bayes net in question is guaranteed to exist and one is asked to find such a Bayes net. We complement our hardness result with a positive result about the sample complexity that is sufficient to recover a parameter-bounded Bayes net that is close (in TV distance) to a given distribution $\mathbb{P}$, represented by some parameter-bounded Bayes net, thereby generalizing a degree-bounded sample complexity result of Brustle et al. (EC 2020).

1. Introduction

Bayesian networks (Pearl, 1988), or simply Bayes nets, are directed acyclic graphs (DAGs), accompanied by a collection of conditional probability distributions (that is, one for each vertex), that are used to represent joint probability distributions over dependent random variables in an elegant and succinct manner. As an example, consider a distribution $\mathbb{P}$ over five Boolean variables $\boldsymbol{X} = \{X_1, X_2, X_3, X_4, X_5\}$. Regardless of the dependencies between the variables, $\mathbb{P}$ can always be represented by a lookup table that has $2^5 - 1 = 31$ entries, with one entry for each possible Boolean outcome except one. However, known dependencies between variables may induce a sparser representation. If X_2 depends on X_1, X_3 depends on $\{X_2, X_5\}$, and X_4 depends on X_3, then the joint distribution $\mathbb{P}(x_1, \ldots, x_5)$ decomposes as

$$\mathbb{P}(x_1) \cdot \mathbb{P}(x_2 \mid x_1) \cdot \mathbb{P}(x_3 \mid x_2, x_5) \cdot \mathbb{P}(x_4 \mid x_3) \cdot \mathbb{P}(x_5).$$

In fact, one can represent $\mathbb{P}$ with a relatively sparse Bayes net $\mathcal{G}$ (see Figure 1) with conditional probability tables (CPTs) associated with each vertex. Observe that $8 < 31$ numbers suffice to describe the CPTs: One for each of the Bernoulli distributions of X_1 and X_5, two for the conditional probability distributions of X_2 and X_4, and four for that of X_3. In the rest of this work, we refer to the numbers used in defining the CPTs above as *parameters*.

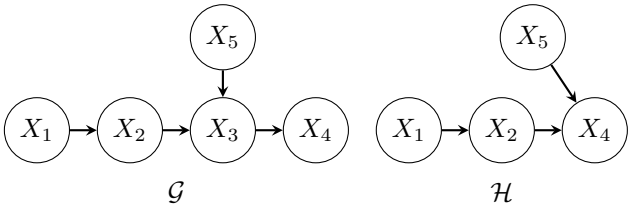

Figure 1. Left: A Bayes net $\mathcal{G}$ such that the distribution $\mathbb{P}$ of our example is represented by $\mathcal{G}$. *Right:* A Bayes net $\mathcal{H}$ such that the distribution that arises from the distribution $\mathbb{P}$ after marginalizing out X_3 is represented by $\mathcal{H}$.

It is a standard result (Pearl, 1988; Chickering et al., 2004) that there exists a Bayes net $\mathcal{G}$ of p parameters that represents a probability distribution $\mathbb{P}$ if and only if $\mathbb{P}$ is Markov with respect to some Bayes net $\mathcal{H}$ of p parameters that has the same underlying DAG as $\mathcal{G}$. (It could be that $\mathcal{G} = \mathcal{H}$, but this is not necessary.) Here, the property of a distribution $\mathbb{P}$ being Markov with respect to a Bayes net $\mathcal{G}$ means that a certain graphical separation condition in the underlying DAG of $\mathcal{G}$, known as d-separation, implies conditional independence in $\mathbb{P}$ (see Section 2.2 for formal definitions). We shall make use of this equivalence in the sequel.

A series of works studied the problem of learning the underlying DAG of a Bayes net from data, by focusing on maximizing certain scoring criterion by the underlying DAG, see, e.g., (Cooper & Herskovits, 1992; Spiegelhalter et al., 1993; Heckerman et al., 1995). This task was later shown to be NP-hard by Chickering (1996), which then raised the following natural fundamental question:

[1]National University of Singapore [2]Indian Institute of Technology Kanpur [3]CNRS@CREATE LTD. Correspondence to: Davin Choo <davin@u.nus.edu>, Dimitrios Myrisiotis <dimitrios.myrisiotis@cnrsatcreate.sg>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria. Copyright 2024 by the author(s).

> Given a succinct description of a distribution $\mathbb{P}$ (that is *not* in terms of a Bayes net), how easy is it to find a Bayes net $\mathcal{G}$ such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$?

Unfortunately, Chickering et al. (2004) showed that deciding whether a given distribution $\mathbb{P}$ is Markov with respect to some Bayes net of at most $p \in \mathbb{N}$ parameters or not is NP-hard.

Remark 1.1. In Chickering et al. (2004), the distribution $\mathbb{P}$ is described as the certain *marginal of a Bayes net* of small in-degree, i.e., a succinct Bayes net description over variables $\boldsymbol{X}$, along with a subset of variables $\boldsymbol{S} \subseteq \boldsymbol{X}$ to marginalize out. For example, any distribution $\mathbb{P}'$ that is Markov with respect to the left Bayes net in Figure 1 is Markov with respect to the right Bayes net in Figure 1 after marginalizing out X_3. Note that all possible distributions over $\boldsymbol{X}$ are Markov with respect to some Bayes net over a clique, but such a Bayes net requires $2^{|\boldsymbol{X}|} - 1$ parameters.

Regarding upper bounds, there are well-known algorithms for learning the underlying DAG of a Bayes net from distributional samples such as the PC (Spirtes et al., 2000) and GES (Chickering, 2002) algorithms. Recently, Brustle et al. (2020) also gave finite sample guarantees of learning Bayes nets (that have n nodes, each taking values over an alphabet Σ) from samples. When given the promise that the underlying DAG has bounded in-degree of d, Brustle et al. (2020, Theorem 10) asserts that using

$$\mathcal{O}\left(\frac{\log \frac{1}{\delta}}{\varepsilon^2}\left(n\,|\Sigma|^{d+1}\log\left(\frac{n|\Sigma|}{\varepsilon}\right) + n \cdot d \cdot \log n\right)\right) \quad (1)$$

samples from the underlying distribution $\mathbb{P}$, one can learn $\mathbb{P}$ up to total variation (TV) distance ε with probability at least $1 - \delta$.

One standard way to measure the complexity of a Bayes net is by imposing an upper bound on any node's in-degree. In this work, we measure the complexity in terms of the number of parameters p, which is a more fine-grained measure than that of maximum in-degree d. For instance, a Bayes net on n Boolean variables with maximum in-degree d could have as few as $O(n + 2^d)$ parameters (e.g., a star graph where d leaves point towards the center of the start) and as many as $\Omega(n \cdot 2^d)$ parameters (e.g., a complete graph).

1.1. Our Contributions

Our two main contributions are that we extend the hardness result of Chickering et al. (2004) and generalize the finite sample complexity result of Brustle et al. (2020).

Contribution 1. We extend the hardness result of Chickering et al. (2004) to the setting where we are guaranteed that the Bayes net in question is promised to have a small number of parameters. In computational complexity theory, this is also known as a *promise problem*, which generalizes a decision problem in that the input is promised to belong to a certain subset of possible inputs. Our new hardness result confirms the common intuition that it is hard to search for a Bayes net $\mathcal{G}$ that is Markov with respect to a given probability distribution, even if it is known that the distribution in question is Markov with respect to a Bayes net that has a small number of parameters.

Definition 1.2 (The `REALIZABLE-LEARN` problem)**.** Given as input variables $\boldsymbol{X} = (X_1, \ldots, X_n)$, a probability distribution $\mathbb{P}$ on $\boldsymbol{X}$, a parameter bound $p \in \mathbb{N}$, and the promise that there exists a Bayes net $\mathcal{G}$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$, output a Bayes net $\mathcal{G}'$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}'$.

Theorem 1.3. *`REALIZABLE-LEARN` is NP-hard.*

Technically speaking, `REALIZABLE-LEARN` is a promise search problem. While NP-hardness results usually revolve around decision problems, NP-hardness naturally extends to the more general case of search problems when Turing reductions are considered. (Turing reductions comprise a very broad class of reductions, whereby an efficient algorithm for a problem yields an efficient algorithm for another.)

Contribution 2. We generalized the finite sample result of Brustle et al. (2020) from the *degree-bounded* setting to the *parameter-bounded* setting.

Theorem 1.4 (Approximating parameter-bounded Bayes nets using samples)**.** *Fix any accuracy parameter $\varepsilon > 0$ and confidence parameter $\delta > 0$. Given sample access to a distribution $\mathbb{P}$ over n variables, each defined on the alphabet Σ, and the promise that $\mathbb{P}$ is Markov with respect to a Bayes net with at most p parameters,*

$$\mathcal{O}\left(\frac{\log \frac{1}{\delta}}{\varepsilon^2}\left(p\log\left(\frac{n\,|\Sigma|}{\varepsilon}\right) + n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}\log n\right)\right)$$

samples from $\mathbb{P}$ suffice to learn the underlying DAG of a Bayes net $\mathcal{G}$ with at most p parameters and define a distribution $\mathbb{Q}$ that is Markov with respect to $\mathcal{G}$ such that $d_{\mathrm{TV}}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon$ with success probability at least $1 - \delta$.

Notice that when all in-degrees are at most d, we have $p \leq (|\Sigma| - 1) \cdot n \cdot |\Sigma|^d$, so our result generalizes the bound of Brustle et al. (2020) given in Equation (1).[1]

Finally, we note that while Theorem 1.4 runs in time polynomial in $1/\delta$ and $1/\varepsilon^2$, and it has exponential dependency on the number of samples from $\mathbb{P}$, similar to Brustle et al. (2020).

[1] One can further generalize Theorem 1.4 to the case where each node has a different alphabet size, e.g., X_i has alphabet Σ_i, but this is a straightforward extension.

1.2. Paper Outline

After preliminaries in Section 2, we give a high-level overview of the techniques behind our results in Section 3. We then formally prove Theorem 1.3 and Theorem 1.4 in Section 4 and Section 5, respectively. Finally, we conclude with an open problem in Section 6.

2. Preliminaries

2.1. Notation

We define the set of natural numbers by $\mathbb{N}$ and all logarithms refer to the natural $\log$.

Distributions are written as $\mathbb{P}$, $\mathbb{Q}$ and graphs in calligraphic letters, e.g., $\mathcal{G}, \mathcal{H}, \mathcal{K}$. For variables/nodes, we use capital letters, small letters for the values taken by them, and boldface versions for a collection of variables, e.g., $X = x$ and $\boldsymbol{X} = \boldsymbol{x}$. As shorthands, we write $[n]$ for $\{1, \ldots, n\}$ and $\mathbb{P}(\boldsymbol{x})$ for $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$. We will often represent the same set of variables of distributions as nodes in a graph.

Problems and algorithms are named in the typewriter font in full caps and capitalized, respectively, e.g., `PROBLEM` and `Algorithm`.

We will also often use Σ to denote the alphabet set of a variable and write $\Delta_{|\Sigma|}$ to denote the corresponding (conditional) probability simplex.

2.2. Graph-Theoretical Notions

Let $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$ be a fully directed graph on $|\boldsymbol{X}| = n$ vertices and $|\boldsymbol{E}|$ edges where adjacencies are denoted with dashes, e.g., $X - Y$, and arc directions are denoted with arrows, e.g., $X \to Y$. For any node $X \in \boldsymbol{X}$, we write $\mathbf{Pa}_{\mathcal{G}}(X) \subseteq \boldsymbol{X}$ to denote its parents and $\mathbf{pa}_{\mathcal{G}}(X)$ to denote the values they take.

A *degree sequence of a graph* $\mathcal{G}$ on vertex set $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ is a list of degrees $\boldsymbol{d} = (d_1, \ldots, d_n)$ of all vertices in the graph, i.e., vertex X_i has degree d_i. A graph $\mathcal{H} = (\boldsymbol{X}, \boldsymbol{E})$ is said to *realize* degree sequence $\boldsymbol{d}$ if the degrees of $\boldsymbol{X}$ in $\mathcal{H}$ agree with $\boldsymbol{d}$. Realizability is defined in a similar fashion for in-degree sequences $\boldsymbol{d}^- = (d_1^-, \ldots, d_n^-)$.

The graph $\mathcal{G}$ is called a *directed acyclic graph (DAG)* if it does not contain any directed cycles and is said to be *complete* if for every two of its nodes $U, V \in \boldsymbol{X}$ either there is an edge $V \to U$ or an edge $U \to V$, i.e., the underlying undirected graph is a clique. A vertex V_i on any simple path $V_1 - \ldots - V_k$ is called a *collider* if the arcs are such that $V_{i-1} \to V_i \leftarrow V_{i+1}$.

A Bayesian network (or Bayes net) $\mathcal{G}$ for a set of n variables $X_1, \ldots, X_n$ is described by a DAG $(\boldsymbol{X}, \boldsymbol{E})$ and n corresponding conditional probability tables (CPTs), e.g., the CPT for $X_i \in \boldsymbol{X}$ describes $\mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i))$ for all possible values of x_i and $\mathbf{pa}_{\mathcal{G}}(X_i)$. The joint distribution for $\mathbb{P}$ factorizes as

$$\mathbb{P}(\boldsymbol{x}) = \prod_{i=1}^{n} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i)),$$

and we say that $\mathcal{G}$ *represents* $\mathbb{P}$.

All independence constraints that hold in the joint distribution of a Bayes net that has underlying DAG $\mathcal{G}$ are exactly captured by the *d-separation* criterion (Pearl, 1988, Section 3.3.1). Two nodes $X, Y \in \boldsymbol{X}$ are said to be *d-separated* in a DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$ given a set $\boldsymbol{Z} \in \boldsymbol{X} \setminus \{X, Y\}$ if and only if there is no $\boldsymbol{Z}$-active path in $\mathcal{G}$ between X and Y; a $\boldsymbol{Z}$-active path is a simple path Q such that any vertex from $\boldsymbol{Z}$ on Q occurs as a collider and any vertex from $\boldsymbol{X} \setminus \boldsymbol{Z}$ appears as a non-collider. Two nodes are *d-connected* if they are not d-separated. It is known that X is d-separated from its non-descendants given its parents (Pearl, 1988, Section 3.3.1, Corollary 4).

A probability distribution $\mathbb{P}$ is said to be *Markov with respect to a DAG* $\mathcal{G}$ if d-separation in $\mathcal{G}$ implies conditional independence in $\mathbb{P}$. Note that any distribution is Markov with respect to the complete DAG, since there are no d-separations implied by this kind of DAG. (Moreover, any Bayes net over a complete DAG requires $2^{|\boldsymbol{X}|} - 1$ parameters to describe.) A probability distribution $\mathbb{P}$ is said to be *Markov with respect to a Bayes net* $\mathcal{G}$ if $\mathbb{P}$ is Markov with respect to the underlying DAG of $\mathcal{G}$.

2.3. Some Problems of Interest

Chickering et al. (2004) introduced a decision problem about learning Bayes nets from data, called `LEARN`, and proved that `LEARN` is NP-hard by showing a reduction from the NP-hard decision problem degree-bounded feedback arc set (`DBFAS`). See Definition 2.1 and Definition 2.2.

Definition 2.1 (The `DBFAS` decision problem)**.** Given a directed graph $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$ with maximum vertex degree of 3, and a positive integer $k \leq |\boldsymbol{E}|$, determine whether there is a subset of edges $\boldsymbol{E}' \subseteq \boldsymbol{E}$ with of size $|\boldsymbol{E}'| \leq k$ such that $\boldsymbol{E}'$ contains at least one directed edge from every directed cycle in $\mathcal{G}$.

Definition 2.2 (The `LEARN` decision problem)**.** Given variables $\boldsymbol{X} = (X_1, \ldots, X_n)$, a probability distribution $\mathbb{P}$ over $\boldsymbol{X}$, and a parameter bound $p \in \mathbb{N}$, determine whether there exists a Bayes net $\mathcal{G}$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$.

In our work, we focus on the particular `LEARN` instances $(\boldsymbol{X}, \mathbb{P}, p)$ used in Chickering et al. (2004), namely `LEARN-DBFAS`.

Definition 2.3 (The LEARN-DBFAS decision problem)**.** Let R denote the reduction of Chickering et al. (2004) from DBFAS to LEARN. We define as LEARN-DBFAS the set of instances of LEARN that are in the range of R.

That is, for any instance I_L of LEARN-DBFAS, there is some instance I_D of DBFAS such that $R(I_D) = I_L$.

An *independence oracle for a distribution* $\mathbb{P}$ is an oracle that can determine, in constant time, whether or not $U \perp\!\!\!\perp V \mid \boldsymbol{Z}$ for any $U, V \in \boldsymbol{X}$ and $\boldsymbol{Z} \subseteq \boldsymbol{X} \setminus \{U, V\}$. We will use the following result by Chickering et al. (2004).

Theorem 2.4 ((Chickering et al., 2004))**.** *LEARN-DBFAS is* NP*-hard even when one is given access to an independence oracle.*

2.4. Selecting a Close Distribution With Finite Samples

The classic method to select an approximate distribution amongst a set of candidate distributions is via the Scheffé tournament of Devroye & Lugosi (2001), which provides a logarithmic dependency on the number of candidates.

In our work, we will use the Scheffé-based algorithm of Daskalakis & Kamath (2014),[2] which given sample access to an input distribution and explicit access to some candidate distributions, outputs with high probability a candidate distribution that is sufficiently close, in total variation (TV) distance (d_{TV}), to the input distribution.

Theorem 2.5 ((Daskalakis & Kamath, 2014))**.** *Fix any accuracy parameter $\varepsilon > 0$ and confidence parameter $\delta > 0$. Suppose there is a distribution $\mathbb{P}$ over variables $\boldsymbol{X}$ and a collection of explicit distributions $\mathbb{Q} = \{\mathbb{Q}_1, \ldots, \mathbb{Q}_m\}$, where each distribution $\mathbb{Q}_i$ is defined over the same set $\boldsymbol{X}$ and there exists some $\mathbb{Q}^* \in \mathbb{Q}$ such that $d_{\mathrm{TV}}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon$. Then, there is an algorithm that uses $\mathcal{O}\left(\frac{\log 1/\delta}{\varepsilon^2} \log m\right)$ samples from $\mathbb{P}$ and returns some $\mathbb{Q} \in \mathbb{Q}$ such that $d_{\mathrm{TV}}(\mathbb{P}, \mathbb{Q}) \leq 10\varepsilon$ with success probability at least $1 - \delta$ and running time $\mathrm{poly}(m, 1/\delta, 1/\varepsilon^2)$.*

To curate a set of candidates $\mathbb{Q}$, we rely on the following lemma of Brustle et al. (2020) which states that any distribution $\mathbb{Q}$ which approximately agrees with $\mathbb{P}$ on the *local* conditional distribution at each node will be close in TV distance to $\mathbb{P}$ on the entire domain.

Lemma 2.6 ((Brustle et al., 2020))**.** *Suppose $\mathbb{P}$ and $\mathbb{Q}$ are Bayes nets on the same DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$ with n nodes. If*

$$d_{\mathrm{TV}}\Big(\mathbb{P}(X \mid \mathbf{Pa}_{\mathcal{G}}(X) = \sigma), \mathbb{Q}(X \mid \mathbf{Pa}_{\mathcal{G}}(X)) = \sigma\Big) \leq \frac{\varepsilon}{n}$$

for all nodes $X \in \boldsymbol{X}$ and possible parent values $\sigma \in \Sigma^{|\mathbf{Pa}_{\mathcal{G}}(X)|}$, then $d_{\mathrm{TV}}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon$.

Although there are infinitely many possible distributions, since we are satisfied with an approximately close distribution, one can discretize the space via an ε-net.

Definition 2.7 (ε-nets; (Vershynin, 2018))**.** Fix a metric space $(\boldsymbol{T}, d)$. For any subset $\boldsymbol{K} \subseteq \boldsymbol{T}$ and $\varepsilon > 0$, a subset $\boldsymbol{N} \subseteq \boldsymbol{K}$ is called an ε-net of $\boldsymbol{K}$ if every point in $\boldsymbol{K}$ is within distance ε to some point in $\boldsymbol{N}$. That is, $\forall x \in \boldsymbol{K}, \exists x_0 \in \boldsymbol{N}$ such that $d(x, x_0) \leq \varepsilon$. We say that $\boldsymbol{N}$ ε-covers $\boldsymbol{K}$.

As we shall see in Section 5, the candidate set $\mathbb{Q}$ will be created by computing an $\frac{\varepsilon}{n}$-net with respect to the TV distance and then applying Lemma 2.6 suitably.

2.5. Other Related Work

We have already referred to some papers that are relevant to our work. We resume this discussion here.

Dasgupta (1999) considers the task of learning the maximum-likelihood polytree from data. The main result of this paper is that the optimal branching (or Chow-Liu tree) is a good approximation to the best polytree. This result is then complemented by the observation that this learning problem is NP-hard, even to approximately solve within some constant factor. Teyssier & Koller (2005) propose a simple heuristic method for addressing the task of learning Bayes nets. Their approach is based on the fact that the best network (of bounded in-degree) consistent with a given node ordering can be found efficiently. Elidan & Gould (2008) present a method for learning Bayes nets of bounded treewidth that employs global structure modifications and that is polynomial both in the size of the graph and the treewidth bound. At the heart of their method is a dynamic triangulation, that they update in a way which facilitates the addition of chain structures that increase the bound on the model's treewidth by at most one. Friedman et al. (2013) introduce an algorithm that achieves learning by restricting the search space. Their iterative algorithm restricts the parents of each variable to belong to a small subset of candidates. They then search for a network that satisfies these constraints and the learned network is then used for selecting better candidates for the next iteration. Ganian & Korchemna (2021) investigate the parameterized complexity of Bayesian Network Structure Learning (BNSL). They show that parameterizing BNSL by the size of a feedback edge set yields fixed-parameter tractability. Kuipers et al. (2022) combine constraint-based methods with greedy or Markov chain Monte Carlo (MCMC) schemes in a method which reduces the complexity of MCMC approaches to that of a constraint-based method.

[2] Their result is actually more general than what we stated here. For instance, they only require sample access to the distributions in $\mathbb{Q} = \{\mathbb{Q}_1, \ldots, \mathbb{Q}_m\}$ while our setting is simpler as we have explicit descriptions of each of these distributions.

3. Technical Overview

Here, we give a brief high-level overview of the techniques used in our results of Theorem 1.3 and Theorem 1.4.

3.1. NP-Hardness of the Realizable Case

By Theorem 2.4, it would suffice to prove that the existence of a polynomial time algorithm for REALIZABLE-LEARN implies that LEARN-DBFAS instances can be solved in polynomial time if one has access to an independence oracle. The desired result will then follow from the facts that we can efficiently (a) compute the number of parameters of a Bayes net and (b) decide whether a given distribution is Markov with respect to a given Bayes net (when given access to an independence oracle).

Suppose we have a polynomial time algorithm Learner for REALIZABLE-LEARN. Note that it is conventional to assume that such an algorithm *always* halts within some polynomial-time bound, and outputs *some* Bayes net, *even when the respective promise is violated*. We define and analyze the following reduction:

> Given an arbitrary instance $(\boldsymbol{X}, \mathbb{P}, p)$ of REALIZABLE-LEARN, run Learner to obtain a Bayes net $\mathcal{G}$. Then check whether $\mathcal{G}$ has at most p parameters and (while using an independence oracle) check whether or not $\mathbb{P}$ is Markov with respect to $\mathcal{G}$. If *both* of these checks are positive, then output YES. Otherwise, output NO. See Section 4 for the formal proof of Theorem 1.3.

3.2. Approximately Learning Parameter-Bounded Bayes Networks

The main idea is to construct an ε-net over all possible DAGs that satisfy the parameter upper bound p, and then apply a well-known bound from the density estimation literature.

For this purpose, we need to count all possible Bayes nets that satisfy the parameter upper bound p. By a counting argument, we see that there are not many possible DAGs that give rise to some Bayes net of at most p parameters. Then, by a counting argument again, we see that there are only a few conditional distributions that are Markov with respect to a Bayes net $\mathcal{G}$ over a DAG that realizes a given in-degree sequence. Thus we are able to bound the number of distributions that cover all possible conditional distributions which are Markov with respect to $\mathcal{G}$. See Section 5 for the formal proof of Theorem 1.4.

4. REALIZABLE-LEARN is NP-hard

To show that REALIZABLE-LEARN is hard, we reduce LEARN-DBFAS to REALIZABLE-LEARN by making polynomially-many calls to an independence oracle. Given any polynomial time algorithm Learner that solves REALIZABLE-LEARN, we will forward the LEARN-DBFAS instance to Learner and examine the produced Bayes net $\mathcal{G}$. We will describe a polynomial time procedure Reduction that uses an independence oracle to determine whether we should correctly output YES or NO for the given LEARN-DBFAS instance. See Figure 2 for a pictorial illustration of our reduction strategy.

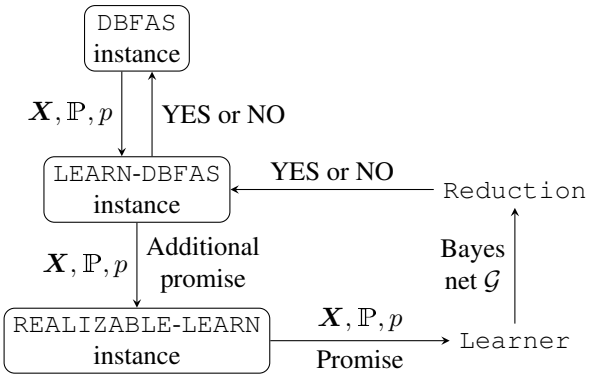

Figure 2. Gavril (1977) showed that DBFAS is NP-hard and Chickering et al. (2004) showed that LEARN-DBFAS is NP-hard, even when given access to an independence oracle for $\mathbb{P}$. REALIZABLE-LEARN is a variant of LEARN-DBFAS with the additional promise that there exists a Bayes net $\mathcal{G}$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$. In this work, we show that *if* one can learn such a Bayes net $\mathcal{G}$ (via some blackbox polynomial time algorithm Learner), then there is a polynomial time algorithm Reduction that correctly answers LEARN-DBFAS. Therefore, REALIZABLE-LEARN is also NP-hard.

We begin by observing that one can easily check the number of parameters of Bayes net given its full description.

Lemma 4.1. *Given a Bayes net over $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$, one can compute the number of its parameters in polynomial time.*

Proof. Let Σ_X denote the alphabet set of $X \in \boldsymbol{X}$. Then, the number of parameters of $\mathcal{G}$ is

$$\sum_{X \in \boldsymbol{X}} \left((|\Sigma_X| - 1) \prod_{U \in \mathbf{Pa}_{\mathcal{G}}(X)} |\Sigma_U| \right),$$

which can be computed in polynomial time. □

The following notion of important edges will come handy in the sequel.

Definition 4.2 (Important edges)**.** Let $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ be a DAG and $\mathbb{P}$ be a distribution over $\boldsymbol{V}$. Then, an edge $e \in \boldsymbol{E}$ is called $(\mathcal{G}, \mathbb{P})$-*important* if $\mathbb{P}$ is Markov with respect to $\mathcal{G}$ but is not Markov with respect to $\mathcal{G}' = (\boldsymbol{V}, \boldsymbol{E} \setminus \{e\})$.

To check whether $\mathbb{P}$ is Markov with respect to $\mathcal{G}$, one could verify that any d-separation in $\mathcal{G}$ implies conditional independence in $\mathbb{P}$. However, this computation seems to be intractable. In contrast, Corollary 4.5 gives a polynomial time algorithm that checks this *while using an independence oracle*.

The correctness of Corollary 4.5 follows from Lemma 4.3 and Lemma 4.4.

Lemma 4.3. *Suppose $\mathbb{P}$ on variables $\boldsymbol{X}$ is Markov with respect to $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$. Then, an edge $A \to B$ in $\boldsymbol{E}$ is not $(\mathcal{G}, \mathbb{P})$-important if $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$.*

Proof. Consider an arbitrary edge $A \to B$ in $\boldsymbol{E}$ such that $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$. Say, $A = X_j$ and $B = X_k$. Letting $\mathcal{G}' = (\boldsymbol{V}, \boldsymbol{E} \setminus (A, B))$ be a subgraph of $\mathcal{G}$ that does not contain the edge $A \to B$, we see that

$$
\begin{aligned}
&\mathbb{P}(\boldsymbol{x}) \\
&= \prod_{i=1}^{n} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i)) && (*) \\
&= \mathbb{P}(x_k \mid \mathbf{pa}_{\mathcal{G}}(X_k)) \cdot \prod_{i \in [n] \setminus k} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i)) \\
&= \mathbb{P}(x_k \mid \mathbf{pa}_{\mathcal{G}}(X_k) \setminus x_j) \cdot \prod_{i \in [n] \setminus k} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i)) && (\dagger) \\
&= \mathbb{P}(x_k \mid \mathbf{pa}_{\mathcal{G}'}(X_k)) \cdot \prod_{i \in [n] \setminus k} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}'}(X_i)) && (\ddagger) \\
&= \prod_{i=1}^{n} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}'}(X_i)),
\end{aligned}
$$

where $(*)$ is due to $\mathbb{P}$ being Markov with respect to $\mathcal{G}$, $(\dagger)$ is due to $X_j \perp\!\!\!\perp X_k \mid \mathbf{Pa}_{\mathcal{G}}(X_k) \setminus \{X_j\}$, and $(\dagger)$ is due to the definition of $\mathcal{G}'$. Since $\mathbb{P}(\boldsymbol{x}) = \prod_{i=1}^{n} \mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}'}(X_i))$, we see that $\mathbb{P}$ is also Markov with respect to $\mathcal{G}'$, and so the edge $A \to B$ is not $(\mathcal{G}, \mathbb{P})$-important. □

Lemma 4.4. *Suppose a distribution $\mathbb{P}$ on variables $\boldsymbol{X}$ is Markov with respect to a DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{E})$. Let $\mathcal{G}' = (\boldsymbol{X}, \boldsymbol{E}')$ be an edge-induced DAG of $\mathcal{G}$ with $\boldsymbol{E}' \subseteq \boldsymbol{E}$. Then, $\mathbb{P}$ is Markov with respect to $\mathcal{G}'$ if and only if $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$ for all edges $A \to B$ in $\boldsymbol{E} \setminus \boldsymbol{E}'$.*

Proof. We prove each direction separately.

(⇐) Suppose that $\mathbb{P}$ is Markov with respect to $\mathcal{G}'$. Consider an arbitrary edge $A \to B \in \boldsymbol{E} \setminus \boldsymbol{E}'$. Since A is an ancestor of B in $\mathcal{G}'$, we have that A remains a non-descendant of B in $\mathcal{G}'$ after removing the edge $A \to B$. So, A and B are d-separated in $\mathcal{G}'$ given $\mathbf{Pa}_{\mathcal{G}'}(B) \setminus \{A\}$, and so $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}'}(B)$ by the Markov property. That is, $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$.

(⇒) Suppose that $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$ for all edges $A \to B$ in $\boldsymbol{E} \setminus \boldsymbol{E}'$. Order the edges in $\boldsymbol{E} \setminus \boldsymbol{E}'$ in an arbitrary sequence, say $e_1, \ldots, e_{|\boldsymbol{E} \setminus \boldsymbol{E}'|}$. Let us remove these edges sequentially, resulting in a sequence of edge-induced DAGs $\mathcal{G} = \mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_{|\boldsymbol{E} \setminus \boldsymbol{E}'|} = \mathcal{G}'$, where $\mathcal{G}_i$ is the edge-induced DAG obtained from removing edges $\{e_1, \ldots, e_i\}$ from $\mathcal{G}$. Observe that non-descendant relationships are preserved as we remove edges, i.e., if A is a non-descendant of B in $\mathcal{G}$, then it is also a non-descendant of B in $\mathcal{G}_i$ for any $i \in \{1, \ldots, |\boldsymbol{E} \setminus \boldsymbol{E}'|\}$. So, we can apply Lemma 4.3 repeatedly: For any $i \in \{1, \ldots, |\boldsymbol{E} \setminus \boldsymbol{E}'|\}$, the edge e_i is not $(\mathcal{G}_{i-1}, \mathbb{P})$-important, so $\mathbb{P}$ is Markov with respect to $\mathcal{G}_i$. That is, $\mathbb{P}$ is Markov with respect to $\mathcal{G}'$. □

Corollary 4.5. *Suppose $\mathbb{P}$ is a distribution over $\boldsymbol{X}$ and $\mathcal{G}$ is a Bayes net over the same set of variables $\boldsymbol{X}$. Then, there is a polynomial time algorithm that uses an independence oracle for $\mathbb{P}$ to decide whether or not $\mathbb{P}$ is Markov with respect to $\mathcal{G}$.*

Proof. Consider the following algorithm `Checker`:

> Given Bayes net $\mathcal{G}$ over a DAG $(\boldsymbol{X}, \boldsymbol{E}')$, consider a DAG $\mathcal{K} = (\boldsymbol{X}, \boldsymbol{E})$, with $\boldsymbol{E}' \subseteq \boldsymbol{E}$, that is a complete supergraph of $(\boldsymbol{X}, \boldsymbol{E}')$. If *every* edge $A \to B \in \boldsymbol{E} \setminus \boldsymbol{E}'$ satisfies $A \perp\!\!\!\perp B \mid \mathbf{Pa}_{\mathcal{G}}(B) \setminus \{A\}$, output YES. Otherwise, output NO.

Note that since any distribution is Markov with respect to the complete DAG (see Section 2.2), $\mathbb{P}$ is Markov with respect to $\mathcal{K}$.

The correctness of `Checker` follows from Lemma 4.4. `Checker` runs in polynomial time as $\mathcal{K}$ can be created in polynomial time with respect to the size of $\boldsymbol{X}$, and the number of edges in $\boldsymbol{E} \subseteq \boldsymbol{E}'$ to check is polynomial in the size of $\boldsymbol{X}$. □

We are now ready to formally prove our first main result.

Theorem 1.3. *`REALIZABLE-LEARN` is* NP*-hard.*

Proof. It suffices to show the existence of a polynomial time algorithm for `REALIZABLE-LEARN` implies that `LEARN-DBFAS` instances can be answered in polynomial time with access to an independence oracle; see Figure 2.

Suppose we have a polynomial time algorithm `Learner` for `REALIZABLE-LEARN`. Let us define a reduction algorithm `Reduction` as follows:

> Given an instance $(\boldsymbol{X}, \mathbb{P}, p)$ of `LEARN-DBFAS`, run `Learner` to obtain a Bayes net $\mathcal{G}$ (see Section 3.1; this is a natural assumption for algorithms solving a search promise problem). Compute the number of parameters of $\mathcal{G}$. Run algorithm `Checker` of Corollary 4.5 on $\mathcal{G}$ to check whether or not $\mathbb{P}$ is Markov with respect to

$\mathcal{G}$. Output YES if $\mathcal{G}$ has at most p parameters *and* $\mathbb{P}$ is Markov with respect to $\mathcal{G}$; else, output NO.

The correctness of `Reduction` follows from the assumption that `Learner` produces a Bayes net $\mathcal{G}$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$ (if the underlying promise is satisfied; otherwise, the output is an arbitrary Bayes net), and the correctness of `Checker`. By assumption, `Learner` is a polynomial time algorithm. By Lemma 4.1, we can compute the number of parameters of $\mathcal{G}$ in polynomial time. By Corollary 4.5, `Checker` is also a polynomial time algorithm. Therefore, the overall running time for `Reduction` is polynomial. □

5. Approximating Bayes Nets

Our strategy for proving our finite sample complexity result (Theorem 1.4) follows that of Brustle et al. (2020, Theorem 10), but we specialize the analysis to the setting where we are given a parameter bound instead of a degree bound. As discussed in Section 1, our result is a generalization of their result since an upper bound on the in-degrees implies a (possibly loose) parameter upper bound.

5.1. Some Graph Counting Arguments

To prove Theorem 1.4, we require an upper bound on the number of possible Bayes nets on n nodes that have at most p parameters (Lemma 5.2). To obtain such a result, we first relate the number of parameters p with a specific given in-degree sequence $(d_1^-, \ldots, d_n^-)$ of a Bayes net, then we upper bound the total number of Bayes nets that has at most p parameters by summing over all suitable in-degree sequences $\boldsymbol{d}^- = (d_1^-, \ldots, d_n^-)$.

Consider an arbitrary Bayes net $\mathcal{G}$ with in-degree sequence $(d_1^-, \ldots, d_n^-)$ and each node taking on $|\Sigma|$ values. Since the conditional distribution for vertex X_i is fully described when we know $\mathbb{P}(x_i \mid \mathbf{pa}_{\mathcal{G}}(X_i))$ for $|\Sigma|-1$ possible values of x_i, with respect to $|\Sigma|^{d_i^-}$ possible values of $\mathbf{pa}_{\mathcal{G}}(X_i)$. Therefore, we see that the Bayes net has

$$\sum_{i=1}^{n} \left((|\Sigma|-1)\, |\Sigma|^{d_i^-} \right) = (|\Sigma|-1) \left(\sum_{i=1}^{n} |\Sigma|^{d_i^-} \right)$$

parameters. Note that this is the exact same reasoning as in Lemma 4.1. So, if the Bayes net has at most p parameters, then

$$\sum_{i=1}^{n} |\Sigma|^{d_i^-} = |\Sigma|^{d_1^-} + \ldots + |\Sigma|^{d_n^-} \leq \frac{p}{|\Sigma|-1}. \tag{2}$$

By the AM-GM inequality, we have that

$$\sum_{i=1}^{n} |\Sigma|^{d_i^-} \geq n \left(\prod_{i=1}^{n} |\Sigma|^{d_i^-} \right)^{\frac{1}{n}} = n|\Sigma|^{\frac{1}{n}\sum_{i=1}^{n} d_i^-}. \tag{3}$$

Combining Equations (2) and (3) together gives us

$$d_1^- + \ldots + d_n^- \leq n \frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}. \tag{4}$$

The following lemma is a combinatorial fact upper bounding on the number of graphs that realize a given degree sequence, which may be of independent interest beyond being used to prove Lemma 5.2.

Lemma 5.1. *Given an in-degree sequence $\boldsymbol{d}^- = (d_1^-, \ldots, d_n^-)$ with non-negative integers $d_1^-, \ldots, d_n^-$, there are at most $\prod_{i=1}^{n} \binom{n-1}{d_i^-}$ DAGs that realize $\boldsymbol{d}^-$.*

Proof. Fix an arbitrary labelling of vertices from X_1 to X_n and consider the sequential process of adding edges into $X_1, \ldots, X_n$. For X_1, there are $\binom{n-1}{d_1^-}$ ways to add d_1^- incoming edges that end at X_1. For X_2, there are $\binom{n-1}{d_2^-}$ possibilities. For X_3, there are *at most* $\binom{n-1}{d_3^-}$ possibilities. Note that some of these choices would be *incompatible* with earlier edge choices as the newly added edges may cause directed cycles to be formed. We repeat this edge adding process until all vertices have added their incoming edges to the graph. So, the *upper bound* is $\prod_{i=1}^{n} \binom{n-1}{d_i^-}$. □

Lemma 5.2. *Suppose that every node takes on at most $|\Sigma|$ values. Then, there are at most*

$$(n-1)^{\frac{n\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}} e^n \left(\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|} + 1 \right)^n$$

possible DAGs over n nodes that may be used to define some Bayes net that has at most p parameters.

Proof. By Lemma 5.1, there are $\prod_{i=1}^{n} \binom{n-1}{d_i^-}$ possible DAGs realizing any fixed in-degree sequence $\boldsymbol{d}^- = (d_1^-, \ldots, d_n^-)$. Let $(*)$ denote the condition that an in-degree sequence $\boldsymbol{d}^-$ yields a graph that has at most p parameters. Then,

$$\begin{aligned}
&\sum_{\boldsymbol{d}^- \text{ satisfies } (*)} \prod_{i=1}^{n} \binom{n-1}{d_i^-} \\
&\leq \sum_{\boldsymbol{d}^- \text{ satisfies } (*)} (n-1)^{d_1^- + \ldots + d_n^-} \\
&\leq (n-1)^{n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}} \sum_{\boldsymbol{d}^- \text{ satisfies } (*)} 1 \\
&\leq (n-1)^{n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}} \binom{n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|} + n}{n} \\
&\leq (n-1)^{n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}} \left(e \left(\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|} + 1 \right) \right)^n
\end{aligned}$$

where the first and last inequalities is because $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k \leq n^k$, the second inequality is due to Equation (4), and the third inequality is obtained via standard "stars and bars" counting. That is, we introduce an auxiliary variable d_0^- and count the number of non-negative integer solutions of

$$d_0^- + d_1^- + \ldots + d_n^- = n \frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}. \qquad \square$$

5.2. Proof of Theorem 1.4

We are now ready to prove Theorem 1.4. Since Lemma 2.6 tells us that it suffices to approximate each local conditional distribution at each node well. So, we will consider an $\frac{\varepsilon}{n}$-net over all such distributions and then apply a tournament style argument (Theorem 2.5) to pick a good candidate amongst the joint distribution obtained by a combination of such candidate local distributions.

Theorem 1.4 (Approximating parameter-bounded Bayes nets using samples)**.** *Fix any accuracy parameter $\varepsilon > 0$ and confidence parameter $\delta > 0$. Given sample access to a distribution $\mathbb{P}$ over n variables, each defined on the alphabet Σ, and the promise that $\mathbb{P}$ is Markov with respect to a Bayes net with at most p parameters,*

$$\mathcal{O}\left(\frac{\log\frac{1}{\delta}}{\varepsilon^2}\left(p\log\left(\frac{n|\Sigma|}{\varepsilon}\right) + n\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}\log n\right)\right)$$

samples from $\mathbb{P}$ suffice to learn the underlying DAG of a Bayes net $\mathcal{G}$ with at most p parameters and define a distribution $\mathbb{Q}$ that is Markov with respect to $\mathcal{G}$ such that $d_{\mathrm{TV}}(\mathbb{P},\mathbb{Q}) \leq \varepsilon$ with success probability at least $1-\delta$.

Proof. Fix a DAG $\mathcal{G}$ satisfying an arbitrary in-degree sequence $\boldsymbol{d}^- = \left(d_1^-, \ldots, d_n^-\right)$. Then, there are $|\Sigma|^{d_1^-} + \ldots + |\Sigma|^{d_n^-}$ local conditional distributions for any Bayes net over $\mathcal{G}$. From Equation (2) above, we know that

$$\sum_{i=1}^{n} |\Sigma|^{d_i^-} = |\Sigma|^{d_1^-} + \ldots + |\Sigma|^{d_n^-} \leq \frac{p}{|\Sigma|-1}.$$

Now, consider an arbitrary local distribution over $k = |\Sigma|$ values and let us upper bound the number of points in an $\frac{\varepsilon}{n}$-net for this metric space. Observe that each possible distribution is essentially an element of the probability simplex Δ_k. To get an $\frac{\varepsilon}{n}$-net of Δ_k, we discretize vectors by rounding them to their nearest multiple of $\frac{\varepsilon}{n|\Sigma|}$. If π is a probability vector, and r_π is its rounding, then $\|\pi - r_\pi\|_1 \leq \frac{\varepsilon}{n|\Sigma|}|\Sigma| = \frac{\varepsilon}{n}$. Therefore the number of discretized vectors is at most $\mathcal{O}\left((n|\Sigma|/\varepsilon)^{|\Sigma|}\right)$.

Therefore, for any fixed DAG $\mathcal{G}$, there is a set of

$$m_1 \in \mathcal{O}\left((n|\Sigma|/\varepsilon)^{\frac{p|\Sigma|}{|\Sigma|-1}}\right) \tag{5}$$

distributions that $\frac{\varepsilon}{n}$-cover any possible joint distributions that can be Markov with respect to a Bayes net over $\mathcal{G}$. Meanwhile, by Lemma 5.2, there are at most

$$(n-1)^{\frac{n\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}} e^n \left(\frac{\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|} + 1\right)^n \tag{6}$$

possible DAGs that may be used to define a Bayes net on n nodes that has at most p parameters. Let m_2 denote this number (of Equation (6)).

We can now define a set of distributions $\mathbb{Q}$ over n variables such that there exists $\mathbb{Q}^* \in \mathbb{Q}$ such that $d_{\mathrm{TV}}(\mathbb{P},\mathbb{Q}) \leq \varepsilon$. Let us denote $m = |\mathbb{Q}|$. Putting together the above bounds, we see that there are at most $m = m_1 \cdot m_2$ candidates suffice, where m_1 and m_2 are from Equations (5) and (6). Therefore, with

$$\mathcal{O}\left(\frac{\log\frac{1}{\delta}}{\varepsilon^2}\log m\right)$$
$$\subseteq \mathcal{O}\left(\frac{\log\frac{1}{\delta}}{\varepsilon^2}\left(p\log\left(\frac{n|\Sigma|}{\varepsilon}\right) + \frac{n\log\left(\frac{p}{n(|\Sigma|-1)}\right)}{\log|\Sigma|}\log n\right)\right)$$

samples from $\mathbb{P}$, Theorem 2.5 chooses a distribution $\mathbb{Q}$ amongst the m candidates such that $d_{\mathrm{TV}}(\mathbb{P},\mathbb{Q}) \leq \varepsilon$ with success probability at least $1-\delta$. $\square$

6. Conclusion

In this work, we showed the hardness result of finding a parameter-bounded Bayes net that represents some distribution $\mathbb{P}$, given sample access to $\mathbb{P}$, *even under the promise that such a Bayes net exists*. On a positive note, we gave a finite sample complexity bound sufficient to produce a Bayes net representing a probability distribution $\mathbb{Q}$ that is close in TV distance to $\mathbb{P}$. Our results generalize earlier known results of Chickering et al. (2004) and Brustle et al. (2020) respectively.

An intriguing open question is as follows:

> Suppose we are given sample access to a distribution $\mathbb{P}$ and are promised that there exists a Bayes net on $\mathcal{G}$ with at most p parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$. Is it hard to find a Bayes net $\mathcal{G}'$ that has $\alpha \cdot p$ parameters such that $\mathbb{P}$ is Markov with respect to $\mathcal{G}'$ (where $\mathcal{G}'$ may not be $\mathcal{G}$), for some constant $\alpha > 1$?

Note that the hardness construction of Chickering et al. (2004) only displayed an *additive* gap in the parameter bound. We conjecture that it is also hard to obtain such a *multiplicative* gap in the parameter bound, even in the promise setting.

Acknowledgements

We would like to thank Dimitris Zoros for helpful discussions. This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-08-013). The work of AB was supported in part by National Research Foundation Singapore under its NRF Fellowship Programme (NRF-NRFFAI-2019-0002) and an Amazon Faculty Research Award. SG's work was partially supported by the SERB CRG Award CRG/2022/007985 and an IIT Kanpur initiation grant. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

Brustle, J., Cai, Y., and Daskalakis, C. Multi-item mechanisms without item-independence: Learnability via robustness. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pp. 715–761, 2020.

Chickering, D. M. Learning Bayesian networks is NP-complete. *Learning from data: Artificial intelligence and statistics V*, pp. 121–130, 1996.

Chickering, D. M. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 (Nov):507–554, 2002.

Chickering, M., Heckerman, D., and Meek, C. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

Cooper, G. F. and Herskovits, E. A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9:309–347, 1992.

Dasgupta, S. Learning polytrees. In Laskey, K. B. and Prade, H. (eds.), *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999*, pp. 134–141. Morgan Kaufmann, 1999.

Daskalakis, C. and Kamath, G. Faster and sample near-optimal algorithms for proper learning mixtures of gaussians. In *Conference on Learning Theory*, pp. 1183–1213. PMLR, 2014.

Devroye, L. and Lugosi, G. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2001.

Elidan, G. and Gould, S. Learning bounded treewidth Bayesian networks. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pp. 417–424. Curran Associates, Inc., 2008.

Friedman, N., Nachman, I., and Pe'er, D. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. *CoRR*, abs/1301.6696, 2013. URL http://arxiv.org/abs/1301.6696.

Ganian, R. and Korchemna, V. The complexity of Bayesian network learning: Revisiting the superstructure. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 430–442, 2021.

Gavril, F. Some NP-complete problems on graphs. In *Proceedings of Conference on Information Sciences and Systems*, pp. 91–95, 1977.

Heckerman, D., Geiger, D., and Chickering, D. M. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243, 1995.

Kuipers, J., Suter, P., and Moffa, G. Efficient sampling and structure learning of Bayesian networks. *J. Comput. Graph. Stat.*, 31(3):639–650, 2022.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., and Cowell, R. G. Bayesian analysis in expert systems. *Statistical science*, pp. 219–247, 1993.

Spirtes, P., Glymour, C. N., and Scheines, R. *Causation, prediction, and search*. MIT press, 2000.

Teyssier, M. and Koller, D. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pp. 548–549. AUAI Press, 2005.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.