

Efficient inference of interventional distributions

Arnab Bhattacharyya
National University of Singapore
arnabb@nus.edu.sg

Sutanu Gayen
National University of Singapore
sutanugayen@gmail.com

Saravanan Kandasamy
Cornell University
sk3277@cornell.edu

Vedant Raval
Indian Institute of Technology Delhi
skved004@gmail.com

N. V. Vinodchandran
University of Nebraska-Lincoln
vinod@cse.unl.edu

May 2021

Abstract

We consider the problem of efficiently inferring interventional distributions in a causal Bayesian network from a finite number of observations. Let \mathcal{P} be a causal model on a set \mathbf{V} of observable variables on a given causal graph G . For sets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$, and setting \mathbf{x} to \mathbf{X} , let $P_{\mathbf{x}}(\mathbf{Y})$ denote the interventional distribution on \mathbf{Y} with respect to an intervention \mathbf{x} to variables \mathbf{X} . Shpitser and Pearl (AAAI 2006), building on the work of Tian and Pearl (AAAI 2001), gave an exact characterization of the class of causal graphs for which the interventional distribution $P_{\mathbf{x}}(\mathbf{Y})$ can be uniquely determined.

We give the first efficient version of the Shpitser-Pearl algorithm. In particular, under natural assumptions, we give a polynomial-time algorithm that on input a causal graph G on observable variables \mathbf{V} , a setting \mathbf{x} of a set $\mathbf{X} \subseteq \mathbf{V}$ of bounded size, outputs succinct descriptions of both an evaluator and a generator for a distribution \hat{P} that is ε -close (in total variation distance) to $P_{\mathbf{x}}(\mathbf{Y})$ where $Y = \mathbf{V} \setminus \mathbf{X}$, if $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable.

We also show that when \mathbf{Y} is an arbitrary set, there is no efficient algorithm that outputs an evaluator of a distribution that is ε -close to $P_{\mathbf{x}}(\mathbf{Y})$ unless all problems that have statistical zero-knowledge proofs, including the Graph Isomorphism problem, have efficient randomized algorithms.

1 Introduction

Density estimation and *parameter learning* are old and classical problems in statistics, studied since the field's inception (e.g., [7, 15, 17]). A more recent focus has been on designing *computationally efficient* learning algorithms, especially in high-dimensional settings. The seminal work of Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [10] set forth the two core computational requirements for distribution learning: (i) evaluation, and (ii) sampling/generation. An **approximate evaluator** for a distribution P takes a domain element \mathbf{v} and outputs the **mass of \hat{P} at \mathbf{v}** , where \hat{P} is a distribution close to P . Similarly, an **approximate generator** for P takes as input a random seed and **outputs an element \mathbf{v} distributed according to \hat{P}** . It is desirable to learn both (approximate) evaluator and generator representations of a distribution, as they can be useful for different downstream inference tasks.

Distribution learning is especially interesting when we cannot directly access the distribution to be learned. One of the most prominent example of such a situation arises in *causal effect estimation*. To estimate the effect of a treatment intervention on some outcome, the “gold standard” is to conduct a randomized experiment where a random sub-population is forcibly treated. However, in many settings (e.g., medicine, economics), it is often not feasible to conduct such experiments, and the only recourse is to make use of observational data. In this work, we study the problem of efficiently learning evaluators and generators for interventional distributions from observational samples.

Our discussion of causality will be in Pearl’s language of *causal Bayesian networks* (or *causal Bayes nets* for short) [13]. A causal Bayes net is a standard *Bayes net that is reinterpreted causally*. Specifically, it makes the assumption of *modularity*: for any variable X , the dependence of X on its parents is an autonomous mechanism that does not change even if other parts of the network are changed. This allows assessment of external interventions, such as those encountered in policy analysis, treatment management, and planning.

We now fix some basic notation for causal Bayes nets to ground the subsequent discussion. The underlying structure of causal Bayes net \mathcal{P} is a directed acyclic graph G . The graph G consists of $n + h$ nodes where n nodes correspond to the *observable* variables \mathbf{V} while the h additional nodes correspond to h *hidden variables* \mathbf{U} . We assume that the observable variables take values over a finite alphabet Σ . Defining the key notion of *c-components* is deferred to Section 2.

The observational distribution P on \mathbf{V} is obtained by interpreting \mathcal{P} as a standard Bayes net over $\mathbf{V} \cup \mathbf{U}$ and marginalizing to \mathbf{V} . An *intervention* is specified by a subset $\mathbf{X} \subseteq \mathbf{V}$ of variables and an assignment¹ $\mathbf{x} \in \Sigma^{|\mathbf{X}|}$. In the interventional distribution, the variables \mathbf{X} are fixed to \mathbf{x} , while each variable $W \in (\mathbf{V} \cup \mathbf{U}) \setminus \mathbf{X}$ is sampled as it would have been in the original Bayes net, according to the conditional distribution $W \mid \mathbf{Pa}(W)$, where $\mathbf{Pa}(W)$ (parents of W) consist of either variables previously sampled in the topological order of G or variables in \mathbf{X} set by the intervention. The marginal of the resulting distribution to \mathbf{V} is the interventional distribution denoted by $P_{\mathbf{x}}$.

The problem of interest for us is to efficiently learn $P_{\mathbf{x}}$, given samples from P . If we leave aside considerations of efficiency, this problem was solved in the important prior works of Shpitser and Pearl [16] and, independently, Huang and Valtorta [8]. We focus here on the work of Shpitser and Pearl. They characterized the set of graphs G for which $P_{\mathbf{x}}$ is not statistically identifiable from P . Moreover when $P_{\mathbf{x}}$ is identifiable, and assuming P is strictly positive, they gave an algorithm that explicitly generates a formula for $P_{\mathbf{x}}$ in terms of P .

If one only has access to samples from P though, Shpitser and Pearl’s algorithm does not guarantee that $P_{\mathbf{x}}$ is learned up to bounded error in total variation distance, either as an evaluator or as a generator, using a polynomial number of samples and time. This is the guarantee which we achieve, when the graph G has bounded in-degree and bounded c-components, the set \mathbf{X} is bounded, and there is no marginalization.

Theorem 1.1 (Informal). *Let \mathcal{P} be a causal Bayes net where the underlying DAG G has in-degree at most d and c-component size at most k . There is an algorithm that given such a causal Bayes net G , a subset $\mathbf{X} \subset \mathbf{V}$ of size at most ℓ such that $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable from P where $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, and ε ; with constant probability, outputs (learns) a distribution \hat{P} as an approximate evaluator and generator so that $d_{\text{TV}}(P_{\mathbf{x}}(\mathbf{Y}), \hat{P}) \leq \varepsilon$. The algorithm uses $m = \tilde{O}\left(\frac{nk^{O(k)}|\Sigma|^{O(k\ell+kd)}}{\alpha^{k\ell}\varepsilon^2}\right)$ samples and $O(m(n + |\Sigma|^{kd+d}))$ time, where α is the minimum probability for any non-empty event defined by the variables in c-components intersecting \mathbf{X} and such c-components’ parents.*

¹Consistent with the convention in the causality literature, we will use a lower case letter (e.g., \mathbf{x}) to denote an assignment to the subset of variables corresponding to its upper case counterpart (e.g., \mathbf{X}).

Note that unlike in [16], we do not require P to be a strictly positive distribution. Indeed if we let α above be a lower bound on $P(\mathbf{v})$, then $\alpha \leq |\Sigma|^{-n}$, and hence, the above sample/time complexity would be exponential. So, it is crucial for efficiency that α be a lower bound on the probability of events defined by only a bounded number of variables.

Theorem 1.1 is proved by showing how to implement the Shpitser-Pearl algorithm while guaranteeing the bound on the distance between the learned and true interventional distributions. This analysis relies on recent work [3, 4] which examined fixed-structure learning of discrete Bayes nets without hidden variables. However, unlike the ‘chain-like’ factorization $P(\mathbf{v}) = \prod_i P(v_i \mid \mathbf{v}_{\mathbf{Pa}(i)})$ of the probability mass function for a Bayes net, the **interventional probability distribution may not admit a nice factorization**; see the examples later in Section 2. Our algorithm exposes a decomposition of $P_{\mathbf{x}}$ into a product of interventional distributions that are either on a small sample space or have ‘chain-like’ factorization, and we show both can be learned efficiently using samples from P . The sample complexity of Theorem 1.1 is nearly optimal in terms of n and ε but it remains an interesting open problem to improve the dependence on d, k, ℓ, α and Σ .

One drawback of Theorem 1.1 is that it considers the interventional distribution on $\mathbf{V} \setminus \mathbf{X}$, whereas the causal effect on only a subset \mathbf{Y} of outcome variables may be relevant and is considered by Shpitser-Pearl [16]. We show that unfortunately, we can not hope for an analog of Theorem 1.1 for an arbitrary subset \mathbf{Y} . In particular, we show that the existence of a learning algorithm that outputs an approximate evaluator representation of $P_{\mathbf{x}}(\mathbf{Y})$ for an arbitrary $\mathbf{Y} \subseteq \mathbf{V}$ will lead to efficient randomized algorithms for all problems in the complexity class SZK which contains hard problems such as the Graph Isomorphism problem. Indeed, we establish the hardness even when the intervention set is empty, i.e., when the goal is to learn the marginal on a subset of variables and when the Bayes net has indegree at most 2.

Theorem 1.2 (Informal). *Suppose there is a randomized polynomial-time algorithm that on input a Bayes net (without hidden variables) distribution P on \mathbf{V} , $\mathbf{Y} \subseteq \mathbf{V}$, and ε ; outputs a representation r of a distribution \hat{P} so that (1) $d_{\text{TV}}(P(\mathbf{Y}), \hat{P}) \leq \varepsilon$, and (2) for every \mathbf{y} , $\hat{P}(\mathbf{y})$ can be evaluated (or even multiplicatively $(1 \pm \varepsilon)$ -approximated) efficiently using r . Then all problems that have statistical zero knowledge (the complexity class SZK), including the Graph Isomorphism problem, can be solved in randomized polynomial time.*

Thus the problem of outputting an evaluator representation of a distribution that approximates the effect of an interventional distribution on an arbitrary subset, even when it is identifiable in the sense of Shpitser-Pearl algorithm, is computationally hard. The formal statement of this theorem along with the proof is presented in Section 4.

Remark 1.3. Using notations of Theorem 1.1, the lower bound of Theorem 1.2 corresponds to the case $d = 2, k = 1, \ell = 0$ and $\alpha = 1$. We also note that for product distributions on n -variables, learning such an approximate evaluator for marginals is trivial.

1.1 Related Work

Pearl [12] introduced Causal Bayesian Networks to formally define interventions and causal effects. Tian and Pearl [19] first studied the problem of identification of causal effects from observations, and gave graphical characterizations for such identifications when the intervention is atomic, i.e. consisting of a single variable and we are interested to determine the effect of this variable on the rest of the variables. They also gave an expression for such an atomic intervention in terms of observational quantities whenever the later is identifiable. Subsequently, Shpitser and Pearl [16] and Huang and Valtorta [8] independently generalized [19] for non-atmoic interventions on any subset of

Algorithm 1: $\text{ID}(\mathbf{y}, \mathbf{x}, P, G)$

Input : Assignments \mathbf{x}, \mathbf{y} , Observational distribution P , ADMG G
Output : $P_{\mathbf{x}}(\mathbf{y})$

- 1 **if** $\mathbf{x} = \emptyset$ **then**
 \perp **return** $\sum_{\mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$
- 2 **if** $\mathbf{V} \setminus \text{An}(\mathbf{Y})_G \neq \emptyset$ **then**
 \perp **return** $\text{ID}(\mathbf{y}, \mathbf{x} \cap \text{An}(\mathbf{Y})_G, \sum_{\mathbf{v} \setminus \text{An}(\mathbf{Y})_G} P, \text{An}(\mathbf{Y})_G)$.
- 3 **if** $\mathbf{W} := (\mathbf{V} \setminus \mathbf{X}) \setminus \text{An}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ *is non-empty* **then**
 \perp For arbitrary assignment \mathbf{w} , **return** $\text{ID}(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.
- 4 **if** $C(G \setminus \mathbf{X} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\})$ **then**
 \perp **return** $\sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_i \text{ID}(s_i, \mathbf{v} \setminus s_i), P, G$.
- 5 **if** $C(G \setminus \mathbf{X}) = \{\mathbf{S}\}$ *is singleton* **then**
 - a) **if** $C(G) = G$ **then**
 \perp **return** FAIL.
 - b) **if** $\mathbf{S} \in C(G)$ **then**
 \perp **return** $\sum_{\mathbf{s} \setminus \mathbf{y}} \prod_{i|V_i \in \mathbf{S}} P(v_i \mid v_1, v_2, \dots, v_{i-1})$.
 - c) **if** $\exists \mathbf{S}' : \mathbf{S} \subset \mathbf{S}' \in C(G)$ **then**
 \perp **return** $\text{ID}(\mathbf{y}, \mathbf{x} \cap \mathbf{S}', \prod_{i|V_i \in \mathbf{S}'} P(V_i \mid (V_1, V_2, \dots, V_{i-1}) \cap \mathbf{S}', (v_1, \dots, v_{i-1}) \setminus \mathbf{S}'), \mathbf{S}')$.

variables, giving graphical characterizations for identifiability and an expression of the causal effect whenever it is identifiable. Recently Barenboim et al. [9] gave estimators for causal effects using weighted regression.

Finite sample bounds for causal inference are being studied only recently. Acharya et al. [1] gave finite bounds for goodness-of-fit testing for non-parametric causal models using observation and experimental data. Bhattacharya et al. [2] gave finite bounds for learning the effect of atomic interventions on the rest of the graph using observational samples, giving finite bounds for [19]. The finite sample and time bounds shown in this paper can be seen as a generalization of [2] for learning non-atomic interventions on the rest of the graph using observational samples. Our work can be seen as a finite-sample version of [16].

Testing various properties of polynomial-time samplable distributions such as uniformity, entropy, and closeness is a fundamental problem that characterizes several zero-knowledge complexity classes [20, 14, 11]. We show our hardness result employing a connection between testing polynomial-time samplable distributions and testing Bayesian networks. Bayesian network is an important statistical model for which finite sample bounds for testing and learning have been given recently [3, 5, 4].

2 Preliminaries

Notation. We use bold and non-bold fonts to denote sets and singleton variables respectively. We use capital and small letters to denote variables and values taken by them respectively.

For two distributions P and Q over Ω , their total variation distance $d_{\text{TV}}(P, Q) = \frac{1}{2} \sum_{i \in \Omega} |P_i - Q_i|$ and their KL distance $d_{\text{KL}}(P, Q) = \sum_{i \in \Omega} P_i \ln \frac{P_i}{Q_i}$. Pinsker's inequality says $d_{\text{TV}}(P, Q) \leq \sqrt{0.5 \cdot d_{\text{KL}}(P, Q)}$.

Definition 2.1. A Bayesian network (or Bayes net in short) P is a distribution over a set of

variables \mathbf{V} defined with respect to a directed acyclic graph G . Each $v \in \mathbf{V}$ takes values from an alphabet Σ . G encodes dependence relationship between the variables: every variable is independent of its non-descendants conditioned on its immediate parents. Hence according to the Bayes rule, P factorizes in the topological order of G such that each $v \in V$ is only conditioned on its parents: $P[\mathbf{V}] = \prod_i P[V_i \mid \mathbf{V}_{\text{parents}(i)}]$.

Causality. We follow Pearl’s formalism of Causal Bayesian Networks [13].

Definition 2.2 (Causal Bayes Net). *A Causal Bayes net P is a Bayes net over the variables $\mathbf{V} \cup \mathbf{U}$ defined with respect to a DAG over $\mathbf{V} \cup \mathbf{U}$. In this DAG, each edge $(X \rightarrow Y)$ denotes a directed causal relationship from X to Y . Only the variables in \mathbf{V} (observed variables) but none from \mathbf{U} (hidden variables) can be observed in the samples from P .*

Given a $X \subseteq V$, and an assignment x to X , the interventional distribution $P_x(V)$, is defined on the DAG where the set of incoming edges to X are removed and X is fixed to x with probability 1. All other variables follow the usual parent-child relation and factorization of P .

It is a standard to assume that the unobservable variables in \mathbf{U} have exactly two observable children. Causal Bayes nets are often represented using a graphical representation over only the observed vertices \mathbf{V} . An Acyclic Directed Mixed Graph (ADMG in short) is such a representation, which consists of a set of variables and two kinds of edges: directed edges E^\rightarrow and bi-directed edges E^\leftrightarrow . The directed edges $(X \rightarrow Y)$ denote parent-child relationship as in a DAG. The bi-directed edges $(X \longleftrightarrow Y)$ denote an indirect correlation between X and Y through a sequence of hidden variables (from U) in between. Thus, the edge set of an ADMG is the union of the directed edges E^\rightarrow and the bidirected edges E^\leftrightarrow . Also for any given set $\mathbf{S} \subseteq \mathbf{V}$, $G_{\overline{\mathbf{S}}}$ denotes the graph obtained from G by removing the incoming edges to \mathbf{S} .

We use Structural Causal Models (SCMs) as defined in [13]. SCMs are defined on Acyclic Directed Mixed Graphs (ADMGs) which contain (i) directed edges between observable variables that form a directed acyclic graph; and (ii) bi-directed edges between observable variables that represent an unobservable confounder between the two variables. For any subset $\mathbf{S} \subseteq \mathbf{V}$, $\mathbf{Pa}^+(\mathbf{S})$ denotes the set of all observable parents of \mathbf{S} (including \mathbf{S}). Let $\mathbf{Pa}^-(\mathbf{S}) = \mathbf{Pa}^+(\mathbf{S}) \setminus \mathbf{S}$.

For ADMGs, the c-component is a key notion that plays a crucial role in the identification of causal effects.

Definition 2.3 (c-component). *An ADMG (or a subgraph of an ADMG) G is a c-component if all the vertices of G are connected by bi-directed edges, or in other words, there exists a subset of bi-directed edges that form a spanning tree of G .*

Definition 2.4 (c-component factorization). *For any ADMG G over observables \mathbf{V} , the c-component factorization is the partition of vertices $\mathbf{V} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m\}$ where the induced subgraphs $G[\mathbf{C}_i]$ s are (maximal) c-components.*

Definition 2.5 (Effective parents of V_i in G). *For any vertex V_i , let \mathbf{C} be the c-component of G that contains V_i . Then the effective parents of V_i is $\mathbf{Pa}^+(\mathbf{C}) \cap \{V_1, V_2, \dots, V_{i-1}\}$.*

Definition 2.6 (α -strong positivity for c-components). *A c-component C is said to be α -strongly positive if for every assignment \mathbf{z} , $P(\mathbf{Pa}^+(\mathbf{C}) = \mathbf{z}) \geq \alpha$.*

We now discuss the identification algorithm of [16] as presented in Algorithm 1. In the remainder of this section, we will go through the different steps of the algorithm by considering two examples.

2.1 Identification Revisited

In this section we discuss the identification algorithm [18] in detail to provide the intuition behind it. Suppose we are given an ADMG G and an observational distribution $\Pr[\mathbf{V}]$. Let $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ be disjoint subsets and \mathbf{x}, \mathbf{y} be assignments to \mathbf{X} and \mathbf{Y} . The goal of the *identification* question is to determine the probability $\Pr[\mathbf{y} \mid do(\mathbf{x})]$. Here we consider the version of the algorithm Algorithm 1 mentioned in [16].

When we are interested in a query of the form $\Pr[\mathbf{y} \mid do(\mathbf{x})]$, certain trivial cases can be handled easily and those cases will correspond to the base cases of their algorithm. The first three steps below will describe those cases.

1. Step 1 handles the base case when \mathbf{X} is an empty set. In this case, the algorithm directly outputs the observational probability $\Pr[\mathbf{y}]$.
2. Step 2 handles the base case when $\Theta := \mathbf{V} \setminus \text{An}(\mathbf{Y})_G$ is non-empty. It is clear by definition that the vertices of Θ is not affecting \mathbf{Y} . Therefore, Θ can be removed from the original graph G .
3. Step 3 handles the base case when $\mathbf{W} := (\mathbf{V} \setminus \mathbf{X}) \setminus \text{An}(\mathbf{Y})_{G_{\mathbf{X}}}$ is non-empty. Suppose \mathbf{W} is non-empty. Since \mathbf{W} does not affect \mathbf{Y} once the vertices \mathbf{X} are intervened, the effect of $do(\mathbf{X})$ on \mathbf{Y} is equivalent to the effect of $do(\mathbf{W} \cup \mathbf{X})$ on \mathbf{Y} . Hence it is convenient to state the question apriori with \mathbf{W} being a subset of \mathbf{X} . This step makes it easier to factorize the required query (based on c-component factorization, Step 4).
4. Suppose $\mathbf{S}_1, \dots, \mathbf{S}_k$ (for $k > 1$) are the c-components of $G[\mathbf{V} \setminus \mathbf{X}]$. Then we get the following formula:

$$\Pr[\mathbf{y} \mid do(\mathbf{x})] = \sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_{i \in [k]} \Pr[\mathbf{s}_i \mid do(\mathbf{v} \setminus \mathbf{s}_i)],$$

where the product term is the resultant of the well-known c-component factorization formula (Definition 2.4) and the summation is the marginalization operation.

5. Suppose the c-component of $G[\mathbf{V} \setminus \mathbf{X}]$ is a singleton $\mathbf{S} = \{\mathbf{V} \setminus \mathbf{X}\}$. Here we have three interesting special cases.
 - (a) Consider the case where the c-component of G is G itself. This results in the existence of hedge (with root set \mathbf{S} and internal nodes \mathbf{X} , and some edges can be removed to make sure each internal node has exactly one outgoing edge. Please refer [16] for the precise definition of Hedge). Hence, it is impossible to uniquely determine the required query.
 - (b) Consider the case where \mathbf{S} is a c-component of G . This means there is no-bidirected edge between \mathbf{S} and $\mathbf{V} \setminus \mathbf{S}$. Due to the absence of bi-directed edges, no backdoor paths exist from \mathbf{X} to \mathbf{S} , resulting in the following formula:

$$\Pr[\mathbf{y} \mid do(\mathbf{x})] = \sum_{\mathbf{v} \setminus \mathbf{y}} \prod_{i \mid V_i \in \mathbf{S}} \Pr[v_i \mid v_1, \dots, v_{i-1}]$$

whose correctness can be easily verified by Bayes rule and marginalization. (Note: One can also apply conditional independence properties of the model on top of the above expression which would remove some of the v_j 's in the conditional terms.)

- (c) This is the final case. Suppose there exists $\mathbf{S}' : \mathbf{S} \subset \mathbf{S}'$ which is a c-component of G . Here, similar to Step 5(b), since there is no bi-directed edge between $\mathbf{X} \setminus \mathbf{S}'$ and the rest of the vertices, it is possible to identify the distribution obtained after the intervention $do(\mathbf{x} \setminus \mathbf{S}')$ - using the following formula:

$$\Pr[\mathbf{S}' \mid do(\mathbf{x} \setminus \mathbf{S}')] = \prod_{V_i \in \mathbf{S}'} \Pr[V_i \mid (\{V_1, \dots, V_{i-1}\}) \cap \mathbf{S}', \mathbf{x} \setminus \mathbf{s}'].$$

where the assignment $\mathbf{x} \setminus \mathbf{s}'$ is the one that's consistent with \mathbf{x} .

The idea here is to first determine $do(\mathbf{x} \setminus \mathbf{S}')$ and then try to recursively determine $do(\mathbf{x} \cap \mathbf{S}')$ on top of that – thus obtaining $do(\mathbf{x})$. Hence the required query is obtained by determining $\Pr[\mathbf{y} \mid do(\mathbf{x} \cap \mathbf{S}')] in the graph $G[\mathbf{S}']$ with observational distribution $\Pr[\mathbf{S}' \mid do(\mathbf{x} \setminus \mathbf{S}')] defined above.$$

Example 1. Consider the ADMG shown in Figure 1a and the identification of $P(y, z_1, z_2 \mid do(x))$ from the observational distribution on this graph. Note that the conditions in steps 1, 2 and 3 of Algorithm 1 do not hold and hence step 4 gets invoked. Since the c-components of $G[\{Z_1, Z_2, Y\} \setminus \{X\}]$ are $\{Y, Z_1\}$ and Z_2 , we get the following formula: $P(y, z_1, z_2 \mid do(x)) = P[y, z_1 \mid do(x, z_2)] \cdot P[z_2 \mid do(x, y, z_1)]$.

For $P[y, z_1 \mid do(x, z_2)]$, the recursive procedure invokes Step 5b to obtain the following equivalent expression $P[z_1 \mid x] \cdot P[y \mid z_1, z_2, x]$.

For the other term $P[z_2 \mid do(x, y, z_1)]$, since Y is not an ancestor of Z_2 , the intervention on Y is not particularly useful. This simplification is taken care by Step 2, which reduces to the following question $P[z_2 \mid do(x, y, z_1)] = P'[z_2 \mid do(x, z_1)]$ on Graph 1b, where $P'[X, Z_1, Z_2] = \sum_y P[X, Z_1, Z_2, y]$. In the next recursive step, the algorithm invokes Step 5c with $\mathbf{S}' = \{Z_2, X\}$, henceforth reduces to a new question $P'[z_2 \mid do(x, z_1)] = P''[z_2 \mid do(x)]$ in Graph 1c, where $P''[Z_2, X] = P'[X]P'[Z_2 \mid X, z_1]$. Finally the algorithm invokes Step 2, obtaining $P''[z_2 \mid do(x)] = P''[z_2] = \sum_x P'[x]P'[z_2 \mid x, z_1]$.

Example 2 Now consider the question of identifying $P[y \mid do(x, r, y)]$ on the ADMG shown in Figure 2a. Note that the conditions in steps 1-4 do not hold for the required query, hence the algorithm directly invokes step 5c where $\mathbf{S}' = \{X, Y, W\}$. This invocation boils down to the identification question of determining $P'[y \mid do(w, x)]$ in Graph 2b, where $P'[W, X, Y] := P[W, X, Y \mid do(r)] = P[W]P[X \mid W, r]P[Y \mid W, X, r]$. In the next recursive call, the algorithm will stop at Step 2 resulting in the identification of $P''[y \mid do(x)]$ in Graph 2c, where $P'' = \sum_w P'[X, Y, w] = \sum_w P[w]P[X \mid w, r]P[Y \mid X, w, r]$. Finally, Step 5b gets invoked in the next call thus resulting in the following expression:

$$P''[y \mid do(x)] = P''[y \mid x] = \frac{\sum_w P[w]P[x \mid w, r]P[y \mid x, w, r]}{\sum_{w, y} P[w]P[x \mid w, r]P[y \mid x, w, r]}.$$

3 Efficient Learning of Interventional Distribution

3.1 Evaluation

In this section we give an algorithm for evaluating the interventional probability $P_{\mathbf{x}}(\mathbf{y})$ for any \mathbf{X} of bounded size and assignments \mathbf{x}, \mathbf{y} to $\mathbf{X}, \mathbf{V} \setminus \mathbf{X}$ respectively. Throughout this section, we will assume $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ and $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable. In that case, the following fact is obvious.

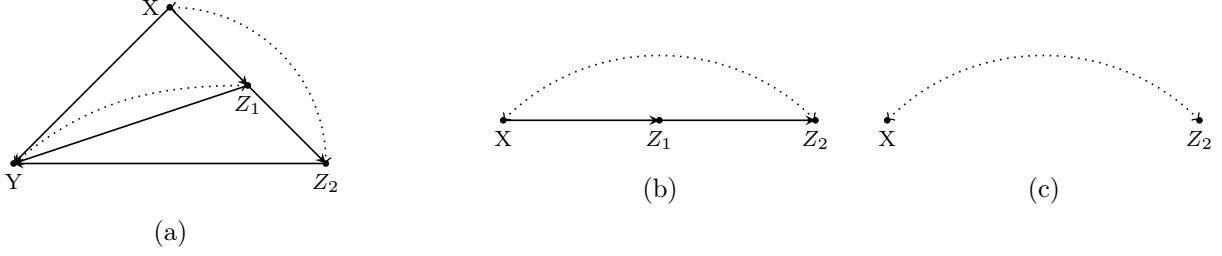


Figure 1

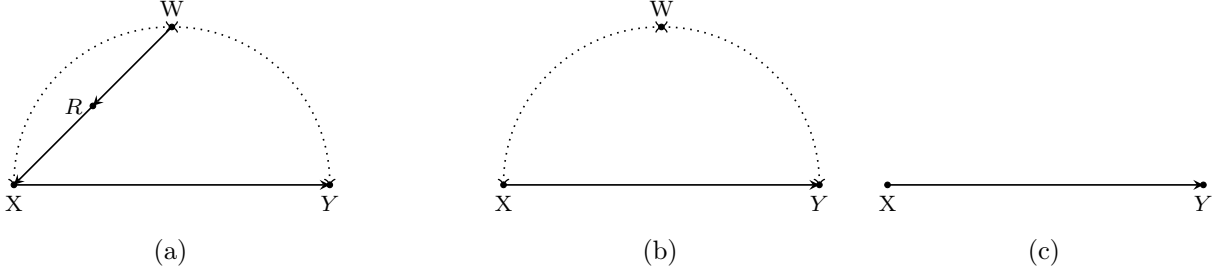


Figure 2

Fact 3.1. *If $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ and $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable, we can remove step 3 and 5a of Algorithm 1, and the summations of steps 4, 1, and 5b, without loss of generality.*

Let $\{\mathbf{C}_i\}_i$ denote the c-component partition of G . Let $\mathbf{X} = \cup_{i=1}^{\ell} \mathbf{X}_i$ be a partition of \mathbf{X} such that $\mathbf{X}_i \subseteq \mathbf{C}_i$, for $i = 1$ to ℓ without loss of generality. We make the following assumption.

Assumption 3.2. $\text{Pa}^+(\mathbf{C}_i)$ is α -strongly positive for every $1 \leq i \leq \ell$.

In this section, we prove the following main theorem.

Theorem 3.3. *We can learn $\hat{P}_{\mathbf{x}}(\mathbf{Y})$ using $m = \tilde{O}\left(\left(\frac{n|\Sigma|^{2k\ell+kd+d}}{\alpha^{k\ell}\varepsilon^2} + \frac{(3k)^{2(k+3)}|\Sigma|^{2k\ell}}{\alpha^2\varepsilon^2}\right) \log \frac{1}{\delta}\right)$ samples from P and in $O(m(n + |\Sigma|^{kd+d}))$ time, such that $d_{\text{TV}}(\hat{P}_{\mathbf{x}}(\mathbf{Y}), P_{\mathbf{x}}(\mathbf{Y})) \leq \varepsilon$ with high probability.*

Define $\mathbf{C}_{>\ell} := \cup_{i>\ell} \mathbf{C}_i$ and $\mathbf{C}_{\leq\ell} := \cup_{i\leq\ell} \mathbf{C}_i$. Let $\mathbf{C}_i \setminus \mathbf{X}_i = \cup_j \mathbf{C}_{ij}$ be the c-component partitions of $G[\mathbf{C}_i \setminus \mathbf{X}_i]$. Then any identifiable joint interventional probability factorizes based on the following claim.

Claim 3.4. *Let $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, $\text{An}(\mathbf{Y}) = \mathbf{V}$, and $P_{\mathbf{x}}(\mathbf{y})$ is identifiable. Then,*

$$P_{\mathbf{x}}(\mathbf{y}) = P(\mathbf{c}_{>\ell} \mid \text{do}(\mathbf{c}_1, \dots, \mathbf{c}_{\ell})) \prod_{i,j} P(\mathbf{c}_{ij} \mid \text{do}(\mathbf{v} \setminus \mathbf{c}_{ij})).$$

Proof. Follows from Step 4 of Algorithm 1. □

Hence it suffices for us to learn $R = \prod_{i,j} P(\mathbf{c}_{ij} \mid \text{do}(\mathbf{v} \setminus \mathbf{c}_{ij})) = P(\mathbf{C}_{\leq\ell} \setminus \mathbf{X} \mid \text{do}(\mathbf{x}, \mathbf{c}_{>\ell}))$ and $Q = P_{\mathbf{c}_{\leq\ell}}(\mathbf{c}_{>\ell}) = P(\mathbf{c}_{>\ell} \mid \text{do}(\mathbf{c}_1, \dots, \mathbf{c}_{\ell}))$. The former distribution is bounded-dimensional and the later is high-dimensional. So, our approach would be to learn R as an explicit p.m.f. table of size at most $|\Sigma|^k$ and Q as a Bayes net, both up to an additive error at most ε .

Now we focus on learning the distribution $R = P(\mathbf{C}_{\leq\ell} \setminus \mathbf{X} \mid \text{do}(\mathbf{x}, \mathbf{c}_{>\ell}))$ for every fixing of $\text{Pa}^-(\mathbf{C}_{\leq\ell} \setminus \mathbf{X})$. $R = \prod_{i,j} R_{i,j}$, where each distribution $R_{i,j} = P[\mathbf{C}_{ij} \mid \text{do}(\mathbf{v} \setminus \mathbf{c}_{ij})]$ starts a recursive

call of Algorithm 1. Each such recursive call forms a recursion tree whose non-leaves correspond recursive calls from steps 2, 4, or 5c, and the leaves correspond to recursive calls from steps 1 or 5b. Our strategy would be to give a $(1 \pm \varepsilon_2)$ -approximate p.m.f. access to the joint distribution for every subsequent recursive call (compared to the true joint distribution), assuming a $(1 \pm \varepsilon_1)$ -approximate p.m.f. access to the joint distribution of the current call. Inductively, we'll get a $(1 \pm \varepsilon_3)$ -approximate access to the leaf distributions: outputs by the leaf calls. Each distribution $R_{i,j}$ is a multiplication of the leaf distributions, possibly with some marginalizations. Marginalization preserves the approximation ratio of the p.m.f. and there are at most k multiplicands, so that our final approximation ratio would be $(1 \pm \varepsilon_3)^k$.

Recursive calls are taken from steps 2, 4, or 5c. In steps 4 and 2, the subsequent call just uses the current distribution (or a marginal of it). In step 5c, the subsequent call uses a very different distribution. Therefore, in a path to the leaf in the recursion tree, we need to give a new joint distribution whenever step 5c is taken. We use the following result to give this distribution, based on whether step 5c was taken for the first time in this root-to-leaf path or not. Let S and D be the two distributions just before and during the recursive call at step 7.

Claim 3.5.

- If $S = P$, we can give a $(1 \pm 3k\varepsilon)$ p.m.f. of \hat{D} for D using $O((kd + d)\alpha^{-2}\varepsilon^{-2} \log \frac{|\Sigma|}{\delta})$ samples and $O((kd + d)|\Sigma|^{kd+d}\alpha^{-2}\varepsilon^{-2} \log \frac{|\Sigma|}{\delta})$ time.
- If $S \neq P$ and we have a $(1 \pm \varepsilon)$ -approximation \hat{Q} for Q , we can give a $(1 \pm 3k\varepsilon)$ -approximate p.m.f. \hat{D} for D using $O(k|\Sigma|^{2k})$ time and no samples.

Proof. Note that each factor of the joint distribution in step 7 is of the form $S[V_i | \mathbf{Z}_i]$, where \mathbf{Z}_i consists of the effective parents of V_i .

In case 1, we take enough samples to empirically learn $S[V_i | \mathbf{z}_i]$ for every fixing of \mathbf{z}_i . Due to α -strong positivity, $\min(S[v_i | \mathbf{z}_i], S[\mathbf{z}_i]) \geq \alpha$ for any v_i, \mathbf{z}_i . Hence from Chernoff's bound, using $O(\alpha^{-2}\varepsilon^{-2} \log \frac{1}{\delta})$ samples the learnt distribution would be point-wise $(1 \pm \varepsilon)$ -close with high probability. Since $|\mathbf{Z}_i| \leq (kd + d)$ our final sample complexity is $m = O((kd + d)\alpha^{-2}\varepsilon^{-2} \log \frac{|\Sigma|}{\delta})$ and the time complexity is $O(m|\Sigma|^{kd+d})$. Since each factor is $(1 \pm \varepsilon)$ -approximate, the approximation for the joint distribution is at most $(1 \pm \varepsilon)^k$.

In case 2, we compute $S[v_i | \mathbf{z}_i] = S[v_i, \mathbf{z}_i]/Q[\mathbf{z}_i]$ by appropriate marginalizations of S , which preserves the approximation. Since step 7 is taken at least once before, the graph-size is at most k . Due to the ratio along with a maximum of k multiplications, the final approximation becomes $\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^k$ -factor. This does not involve sampling and can be done in $O(k|\Sigma|^{2k})$ time, since in this case Q must be over at most k variables.

In either case, we return \hat{D} as a p.m.f. table of size $|\Sigma|^k$. □

Similarly for the leaf calls from step 1 or 5b, we approximate their (terminal) output distributions (henceforth referred to as leaf distributions) by marginalization or conditional sampling, depending on whether the current distribution is P or not. Inductively, we would get an approximation guarantee between the true and estimated leaf distributions. Our output just consists of p.m.f. tables for each joint distribution \hat{S} just before a leaf call. In that case, each leaf distribution \hat{D} is simply a marginalization (step 1) or product of conditional probabilities (step 6) of \hat{S} .

We now consider the leaf calls from step 1 or 5b. Again, we split into two cases, depending on whether the leaf call was taken using the original distribution P or not. Let D be the distribution of the leaf call.

Claim 3.6. *Let D be a true leaf distribution in the recursion tree. Then our corresponding distribution \hat{D} as mentioned above is point-wise $(1 \pm (3k)^k\varepsilon)$ -approximate for D .*

Proof. If $D = P$ we get a sampling access to P . If $D \neq P$, we get D as a p.m.f. table.

Step 1 returns the joint distribution over \mathbf{Y} , over at most $|\Sigma|^k$ items. If $D = P$, we learn this distribution up to point-wise $(1 \pm \varepsilon)$ -factor using $\tilde{O}(k\alpha^{-1}\varepsilon^{-2})$ samples with high probability. If $D \neq P$, D itself would be the output distribution. In either case, we return a p.m.f. table of size at most $|\Sigma|^k$.

If step 5c is taken with $D = P$, we need learn each $P[v_i | \mathbf{z}_i]$, where \mathbf{z}_i is the effective parent set of V_i , and is of size at most $(kd + d)$. So, we iterate through all possible \mathbf{z}_i and learn each term $P[v_i | \mathbf{z}_i]$ point-wise up to $(1 \pm \varepsilon)$ -error by rejection sampling with high probability. There are at most k such terms, so that the sample complexity would be $m = \tilde{O}((kd + d)\alpha^{-2}\varepsilon^{-2})$ and the time complexity would be $O(m|\Sigma|^{kd+d})$.

If step 5c is taken with $D \neq P$, then it must be preceded by a call from step 5b, since only that step changes the distribution by our construction. Then D is a distribution over at most k variables. We obtain all possible $D[v_i | \mathbf{z}_i] = D[v_i \circ \mathbf{z}_i] / D[\mathbf{z}_i]$ terms by appropriate marginalizations of D using $\tilde{O}(|\Sigma|^{2k})$ time and no samples.

In both the above cases of step 5c, we return all possible conditional probability tables required for evaluating the output formula of this step. Note that the recursion depth is at most k and we lose a factor of $(1 \pm 3k\varepsilon)$ from $(1 \pm \varepsilon)$ in each depth. Therefore, the approximation ratio for the leaf distributions is at most $(1 \pm (3k)^k\varepsilon)$. \square

Each $R_{i,j}$ is a product of several leaf distributions. Our final output distribution $\hat{R}_{i,j}$ just uses \hat{S} in place of each such leaf distribution S . Since $R = \prod_{i,j} R_{i,j}$ is a product of at most $k\ell$ leaf distributions, the following claim is obvious.

Claim 3.7. \hat{R} is point-wise $(1 \pm (3k)^{k+1}\ell\varepsilon)$ -approximate for R .

We next focus on learning the distribution Q .

Lemma 3.8. Let $\mathbf{Pa}^+(\mathbf{C}_i)$ be α -strongly positive. Then for every fixing of $\mathbf{C}_{\leq \ell}$, Q can be learnt as a Bayes net \hat{Q} of indegree at most $(kd + d)$ using $m = \tilde{\Theta}\left(\frac{n|\Sigma|^{kd+d}}{\alpha^{|\mathbf{C}_{\leq \ell}|}\varepsilon} \log \frac{n|\Sigma|^{kd+d+1}}{\delta}\right)$ samples and $O(mn)$ time such that $d_{\text{KL}}(Q, \hat{Q}) \leq \varepsilon$ with probability at least $(1 - \delta)$.

We will now introduce some of the tools used in proving Lemma 3.8. The following Claim relates the p.m.f.s of Q and P . Here we used Tian's factorization to write $Q = \prod_{V_i \in \mathbf{C}_{>\ell}} P(V_i | \mathbf{Z}_i)$, where \mathbf{Z}_i is the effective parents of V_i .

Claim 3.9. Let \mathbf{v} be an assignment to \mathbf{V} and \mathbf{w} be its restriction to $\mathbf{C}_{>\ell}$. Then $\alpha^{|\mathbf{C}_{\leq \ell}|} \leq \frac{P[\mathbf{v}]}{Q[\mathbf{w}]} \leq 1$.

Proof. $\frac{P[\mathbf{v}]}{Q[\mathbf{w}]} = \frac{\prod_{V_i \in \mathbf{V}} P(v_i | \mathbf{z}_i)}{\prod_{V_i \in \mathbf{C}_{>\ell}} P(v_i | \mathbf{z}_i)} = \prod_{V_i \in \mathbf{C}_{\leq \ell}} P(v_i | \mathbf{z}_i) \geq \prod_{V_i \in \mathbf{C}_{\leq \ell}} P(v_i \circ \mathbf{z}_i) \geq \alpha^{|\mathbf{C}_{\leq \ell}|}$. \square

We closely follow the d_{KL} -learning result for Bayes nets given in [4]. Let $\mathbf{Z}'_i = \mathbf{Z}_i \setminus \mathbf{C}_{\leq \ell}$. Our algorithm just learns the add-1 empirical distribution $\hat{P}[v_i | \mathbf{Z}'_i = \mathbf{a}]$ on the conditional samples from $P[v_i | \mathbf{Z}'_i = \mathbf{a}]$. Our learnt distribution \hat{Q} consists of the $\hat{P}[v_i | \mathbf{Z}'_i = \mathbf{a}]$'s, in place of every $P[v_i | \mathbf{Z}'_i = \mathbf{a}]$ in the Bayes net factorization for Q .

Fact 3.10 ([6, 5, 4]).

$$d_{\text{KL}}(Q, \hat{Q}) = \sum_{i \in \mathbf{C}_{>\ell}} \sum_{\mathbf{a}} Q[\mathbf{Z}'_i = \mathbf{a}] d_{\text{KL}}(P[v_i | \mathbf{Z}'_i = \mathbf{a}], \hat{P}[v_i | \mathbf{Z}'_i = \mathbf{a}]).$$

We also have the following guarantee about the add-1 estimator.

Fact 3.11 ([4]). Let D be an unknown distribution over k items and \hat{D} be its add-1 empirical distribution of m samples. Then if $m \gtrsim \frac{k}{\varepsilon} \log \frac{k}{\delta} (\log \frac{k}{\varepsilon} + \log \log \frac{k}{\delta})$ then $d_{\text{KL}}(D, \hat{D}) \leq \varepsilon$ with probability at least $(1 - \delta)$.

We are now ready to prove Lemma 3.8.

Proof of Lemma 3.8. We analyze the two cases: $Q[\mathbf{Z}'_i = \mathbf{a}] \geq \frac{\varepsilon}{n|\Sigma|^{kd+d} \log(m+|\Sigma|)}$ and otherwise.

In the former case, $P[\mathbf{Z}'_i = \mathbf{a}] \geq \frac{\alpha^{|\mathbf{C}_{\leq \ell}|} \varepsilon}{n|\Sigma|^{kd+d+1}}$ from Claim 3.9 and $m = \tilde{\Theta}\left(\frac{n|\Sigma|^{kd+d}}{\alpha^{|\mathbf{C}_{\leq \ell}|} \varepsilon} \log \frac{n|\Sigma|^{kd+d+1}}{\delta}\right)$ samples would ensure at least $\tilde{\Theta}\left(\frac{n|\Sigma|^{kd+d+1} \cdot Q[\mathbf{Z}'_i = \mathbf{a}]}{\varepsilon} \log \frac{n|\Sigma|^{kd+d+1}}{\delta}\right)$ conditional samples are seen from $P[v_i | \mathbf{Z}'_i = \mathbf{a}]$ with high probability from Chernoff's bound. Hence, $d_{\text{KL}}(P[v_i | \mathbf{Z}'_i = \mathbf{a}], \hat{P}[v_i | \mathbf{Z}'_i = \mathbf{a}]) \leq \frac{\varepsilon}{n|\Sigma|^{kd+d} Q[\mathbf{Z}'_i = \mathbf{a}]}$ from Fact 3.11 except with probability at most $\frac{\varepsilon}{n|\Sigma|^{kd+d}}$.

In the later case, $Q[\mathbf{Z}'_i = \mathbf{a}] \leq \frac{\varepsilon}{n|\Sigma|^{kd+d} \log(m+|\Sigma|)}$ and $d_{\text{KL}}(P[v_i | \mathbf{Z}'_i = \mathbf{a}], \hat{P}[v_i | \mathbf{Z}'_i = \mathbf{a}]) \leq \log(m + |\Sigma|)$, since the least add-1 probability of at most m conditional samples is $\frac{1}{m+|\Sigma|}$.

Combining all the cases, the summation of the RHS of Fact 3.10 evaluates to at most $O(\varepsilon)$. \square

Now we describe the overall construction of the evaluator for $P_{\mathbf{x}}(\mathbf{Y}) = Q \cdot R$, where $Q = \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i)$, $R = \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i)$, \mathbf{Z}_i is the conditioning set for V_i , Q is the Bayes net and S_i 's are the leaf distributions defined before. This is very similar to a Bayes net factorization except that the probability distribution for the factors need not be the same. Our evaluator is

$$\hat{P}_{\mathbf{x}}(\mathbf{y}) = \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i), \quad (1)$$

where \hat{Q} comes from Lemma 3.8 and \hat{S}_i comes from R in Claim 3.7.

Proof of Theorem 3.3. We get $d_{\text{TV}}(\prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i), \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}(V_i | \mathbf{Z}_i)) \leq \varepsilon$ for any fixed $\mathbf{c}_{\leq \ell}$ from Lemma 3.8 and Pinsker's inequality, using $m = \tilde{\Theta}\left(\frac{n|\Sigma|^{kd+d}}{\alpha^{|\mathbf{C}_{\leq \ell}|} \varepsilon^2} \log \frac{n|\Sigma|^{kd+d+1}}{\delta}\right)$ samples and $O(mn)$ time with high probability.

We scale down ε by a $(3k)^{k+1} \cdot \ell$ factor to get that for any fixed $\mathbf{c}_{>\ell}$, \hat{R} is point-wise $(1 \pm \varepsilon)$ -approximate for R with high probability using $m = O((3k)^{2(k+3)} \ell^3 d \alpha^{-2} \varepsilon^{-2} \log \frac{|\Sigma|^{k\ell}}{\delta})$ samples and $O(m|\Sigma|^{kd+d})$ time from Claim 3.5 and Claim 3.7.

Combining the above two pieces, we get at most $\varepsilon(|\Sigma|^{k\ell} + 1)$ error as follows. We get the theorem by an appropriate scaling.

$$\begin{aligned} & d_{\text{TV}}(P_{\mathbf{x}}(\mathbf{y}), \hat{P}_{\mathbf{x}}(\mathbf{y})) \\ &= \sum_{\mathbf{c}_{\leq \ell}, \mathbf{c}_{>\ell}} \left| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i) - \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i) \right| \\ &= \sum_{\mathbf{c}_{\leq \ell}, \mathbf{c}_{>\ell}} \left| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i) - \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i) \right. \\ &\quad \left. + \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i) - \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i) \right| \\ &\leq \sum_{\mathbf{c}_{>\ell}} \sum_{\mathbf{c}_{\leq \ell}} \left| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i) - (1 \pm \varepsilon) \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i) \right| \end{aligned}$$

$$\begin{aligned}
& + \sum_{\mathbf{c} \leq \ell} \sum_{\mathbf{c} > \ell} \left| \prod_{V_i \in \mathbf{C}_{>\ell}} Q_i(V_i | \mathbf{Z}_i) - \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}_i(V_i | \mathbf{Z}_i) \right| \\
& \leq \varepsilon \sum_{\mathbf{c} > \ell} \sum_{\mathbf{c} \leq \ell} \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i | \mathbf{Z}_i) + \sum_{\mathbf{c} \leq \ell} \varepsilon \leq \varepsilon(1 + |\Sigma|^{k\ell}). \quad \square
\end{aligned}$$

3.2 Generator

In this section, we give an algorithm for generating samples approximately according to the distribution $P_{\mathbf{x}}(\mathbf{y})$. Our algorithm generates samples according to the distribution $\hat{P}_{\mathbf{x}}(\mathbf{y})$ given in (1).

Theorem 3.12. *We can generate independent samples from $\hat{P}_{\mathbf{x}}(\mathbf{y})$ in $O(n)$ time.*

Proof. We have $P_{\mathbf{x}}(\mathbf{y}) = \prod_{V_i \in \mathbf{C}_{>\ell}} \hat{Q}(V_i | \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \hat{S}_i(V_i | \mathbf{Z}_i)$ from (1), where \hat{Q} comes from Lemma 3.8 and \hat{S} comes from R of Claim 3.7. Note that either in Lemma 2 or Claim 8, the effective graph in any step of the recursion is always a subgraph of G . Hence, the topological order between V_i and \mathbf{Z}_i is never violated. In particular, we can sample each random variable of \mathbf{Y} in the topological order of $\mathbf{V} \setminus \mathbf{X}$. Then at any step of this sampling, whenever we try to sample some $v_i \sim \hat{Q}(V_i | \mathbf{Z}_i)$ or $\sim \hat{S}_i(V_i | \mathbf{Z}_i)$, \mathbf{Z}_i would always be sampled before it and assigned. \square

Remark 3.13. It follows that we can also sample from any $\mathbf{T} \subseteq \mathbf{V} \setminus \mathbf{X}$, ignoring the unnecessary variables. However, evaluating the marginal on \mathbf{T} in general remains computationally expensive.

3.3 Examples

In this section, we illustrate our evaluator and generator with the help of two simple examples.

Example 1: Consider the graph in Figure 1a taken from [18]. We would like to learn the intervention $P_x(z_1, z_2, y)$. The following sequence of steps are taken in our algorithm.

1. The starting graph has two c-components (X, Z_2) and (Y, Z_1) . Only (X, Z_2) contains an intervening variable. So, we'll assume (X, Z_1, Z_2) is α -strongly positive.
2. Step 4 is taken first and generates: $\text{ID}((y, z_1), (x, z_2), P, G) * \text{ID}(z_2, (x, y, z_1), P, G)$. Since, P is unchanged, we don't do anything.
3. The first ID takes step 6 and generates the formula: $P[y | x, z_1, z_2]P[z_1 | x]$. Note that we are learning this distribution jointly as a Bayes net, assuming α -strong positivity of only (X, Z_1, Z_2) .
4. Y is marginalized out in step 2 from the second ID. Hence, we don't change the distribution. Next call is $\text{ID}(z_2, (x, z_1), P[X, Z_1, Z_2], H)$, where H is the graph of Figure 1b.
5. Step 7 is taken next: $\text{ID}(z_2, x, S[X', Z_2], T)$, where $S[X', Z_2] = P[X', Z_2 | \text{do}(z_1)] = P[X']P[Z_2 | z_1, X']$. We learn S up to point-wise $(1 \pm 6\varepsilon)$ -factor using $\tilde{O}(\alpha^{-2}\varepsilon^{-2})$ samples and store it as a table. T is the graph shown in Figure 1c.
6. Step 2 next prunes X from T . The distribution is a marginal and hence S is unchanged.
7. Step 1 generates the formula: $S[z_2]$.

At this point, we will be able to evaluate and sample $P[y | x, z_1, z_2]P[z_1 | x]S[z_2]$ using our Bayes net P and the explicit table for S .

Example 2: Consider the graph in Figure 2a taken from [9]. We are interested to learn the intervention $P_{(w,r,y)}(Y)$. The following sequence of steps are taken in our algorithm.

1. The starting graph G has two c-components: (W, X, Y) and R . Both these components contain an intervening variable and hence assumed α -strongly positive together with their parents.
2. First, step 7 is taken: $\text{ID}(y, (w, x), Q[W', X', Y], H)$, where $Q[W', X', Y] = P[W'X'Y \mid \text{do}(r)] = P[W']P[X' \perp W', r]P[Y \mid X', r, W']$. So, we learn Q as a point-wise $(1 \pm 9\varepsilon)$ -approx. p.m.f. table using $\tilde{O}(\alpha^{-2}\varepsilon^{-2})$ samples from P . The graph H is shown in Figure 2b.
3. Next, step 2 prunes W from H . The new distribution is the marginal of Q on X', Y and so Q 's table is passed on unchanged. The new graph T is shown in Figure 2c. So, the next call is $\text{ID}(y, x, Q[X', Y], T)$.
4. Finally, step 6 is taken, which generates the final formula $Q[y \mid x]$.

At this point, we will be able to evaluate and sample $Q[y \mid x]$ using our stored table for Q .

4 Hardness of Evaluating Marginals

In this section we show the computational hardness for approximately giving a succinct evaluator for the *marginal* of an interventional distribution in general. This is in contrast to our result in Section 3.1, where we showed for the specific case that whenever \mathbf{X} is of bounded size and $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, we can efficiently give a generator as well as an evaluator for $\hat{P}_{\mathbf{x}}(\mathbf{Y})$, which was ε close to $P_{\mathbf{x}}(\mathbf{Y})$ in d_{TV} . Thus, our hardness in this section applies to the case when $\mathbf{Y} \subset \mathbf{V} \setminus \mathbf{X}$. In fact we show the hardness for the empty intervention $P(\mathbf{Y})$, which is just the marginal of a completely observed Bayesian network of indegree 2.

We need the following definitions.

Definition 4.1 (Polynomial-time samplable distributions). *Given a Boolean circuit C_n mapping n bits to m bits, the distribution sampled by C_n is obtained by uniformly choosing $x \in \{0, 1\}^n$ and evaluating C on x . A distribution is polynomial-time samplable distribution if it is sampled by a circuit C_n of size $\text{poly}(n)$. We often use C itself to denote the distribution sampled by the circuit C_n .*

We will use the following hardness result for the tolerant testing of two polynomial-time samplable distributions.

Definition 4.2 (Tolerant testing of polynomial-time samplable distributions). *Let DISTCKT be the following problem: given the encodings of two boolean circuits C_n and D_n , both of which output exactly $m = \text{poly}(n)$ bits, distinguish between the two cases: $d_{\text{TV}}(C_n, D_n) \leq 1/3$ versus $d_{\text{TV}}(C_n, D_n) \geq 2/3$.*

Theorem 4.3 ([14]). *DISTCKT is complete for the complexity class Statistical Zero Knowledge (denoted SZK).*

The class SZK contains several hard computational problems including the Graph Isomorphism problem and is believed to be computationally harder than BPP; the class of problems that admit efficient randomized algorithms.

In the following, we show that DISTCKT reduces to the problem of computing an efficient, approximate evaluator for the marginal distribution of a Bayes net, thereby showing that the later problem is hard. We formally define this problem now.

Definition 4.4 (Efficient ε -approximate evaluator circuit). A circuit $\mathcal{E}_P : \{0,1\}^n \rightarrow \mathbb{R}$ is called an efficient ε -additive evaluator for a distribution P over $\{0,1\}^n$, if there exists a distribution \hat{P} over $\{0,1\}^n$ such that:

1. [Additive approximation] $d_{\text{TV}}(P, \hat{P}) \leq \varepsilon$,
2. [Evaluation query] For any $x \in \{0,1\}^n$, \mathcal{E}_P on input x outputs a number $p \in [1 - \varepsilon, 1 + \varepsilon]\hat{P}(x)$ in $\text{poly}(n)$ time,

Definition 4.5. Let BAYESMARGINAL be the following problem: given a parameter ε , samples from an unknown Bayes net P , and a $S \subseteq [n]$, output an efficient ε -approximate evaluator circuit $Q : \{0,1\}^{|S|} \rightarrow \mathbb{R}$ for the marginal distribution of P over S .

We also need the following result for approximating the variation distance additively.

Theorem 4.6 ([3]). Let P and Q be two unknown distributions over Ω . Then given access to two efficient ε -approximate evaluator circuits \mathcal{E}_P and \mathcal{E}_Q for P and Q , we can estimate $d_{\text{TV}}(P, Q)$ up to an additive 4ε error with $(1 - \delta)$ probability using $O(\varepsilon^{-2} \log \frac{1}{\delta})$ samples from P and $O(\varepsilon^{-2} \log \frac{1}{\delta})$ evaluation queries to \mathcal{E}_P and \mathcal{E}_Q .

Theorem 4.7. If BAYESMARGINAL has a randomized polynomial-time algorithm even for Bayes nets of indegree at most 2, then DISTCKT \in BPP and hence SZK \subseteq BPP.

Proof. Let \mathcal{A} be the hypothetical randomized polynomial-time algorithm for BAYESMARGINAL. We solve DISTCKT on C_n and D_n using \mathcal{A} as follows.

Firstly, we assume without loss of generality all the AND, OR, NOT gates of C_n and D_n have at most 2-arguments, since otherwise we stack such gates, increasing the circuit size by at most a polynomial. Then, the two circuits are exactly Bayes nets of indegree at most 2, whose source nodes are random and intermediate nodes follow deterministic functions. We also denote by C_n and D_n these two samplable distributions with a slight abuse of notation. Let S_C and S_D denote the sets of output bits of C_n and D_n , such that $|S_C| = |S_D|$. Let C_{all} and D_{all} denote the joint distributions of *all the gates* of C_n and D_n . Hence, C_n and D_n are respectively the marginals of C_{all} and D_{all} over the sets S_C and S_D .

We run \mathcal{A} on $(C_{\text{all}}, S_C, 1/40)$ to get an efficient ε -approximate evaluator \mathcal{E}_C for C_n with high probability. Similarly we get \mathcal{E}_D for D_n . Moreover, C_n can be sampled in randomized polynomial time. Hence from Theorem 4.6, using \mathcal{E}_C , \mathcal{E}_D , and samples from C_n , we can approximate $d_{\text{TV}}(C_n, D_n)$ up to an additive error $1/10$ in polynomial-time with high probability. This would show DISTCKT \in BPP. \square

Finally, we show a stronger hardness result for getting an evaluator for the marginal of a Bayes net if we want a *multiplicative* approximation of the p.m.f., formally defined below.

Definition 4.8 (Efficient c -multiplicative evaluator circuit). A circuit $\mathcal{E}_P : \{0,1\}^n \rightarrow \mathbb{R}$ is called an efficient c -multiplicative evaluator circuit for a distribution P over $\{0,1\}^n$, if there exists a distribution \hat{P} over $\{0,1\}^n$ such that:

1. [Multiplicative approximation] $\hat{P}(x)/P(x) \in [1/c, c]$ for some constant $c > 1$ and for any $x \in \{0,1\}^n$.
2. [Evaluation] for any $x \in \{0,1\}^n$, \mathcal{E}_P on input x outputs $\hat{P}(x)$ in $\text{poly}(n)$ time.

Definition 4.9 (BAYESMARGINALMULT). *Let BAYESMARGINALMULT be the following problem: given a Bayesian network P over the binary variables $[n]$, and a $S \subseteq [n]$, return an efficient c -multiplicative evaluator circuit for the marginal distribution of P over S .*

We reduce from the circuit evaluation problem which is well-known to be NP-hard.

Definition 4.10 (CIRCEVAL). *Given the encoding of a Boolean circuit $C : \{0, 1\}^n \rightarrow \{0, 1\}$ as input, decide whether there exists an x such that $C(x) = 1$ or not.*

Theorem 4.11. *CIRCEVAL is NP-complete.*

Now, we give a reduction from CIRCEVAL to BAYESMARGINALMULT, showing that the later problem is unlikely to be in randomized polynomial-time.

Theorem 4.12. *If BAYESMARGINALMULT has a randomized polynomial time algorithm even for Bayes nets of indegree at most 2, then CIRCEVAL \in BPP and hence $\text{NP} \subseteq \text{BPP}$.*

Proof. Let \mathcal{A} be a hypothetical randomized polynomial time algorithm for BAYESMARGINALMULT. Let $C : \{0, 1\}^n \rightarrow \{0, 1\}$ be the instance of the CIRCEVAL problem. Let b denote the output bit of C . We also denote by C , the joint distribution of all its gates when its input bits are chosen randomly.

As argued in the proof of Theorem 4.7, C is a Bayes net of indegree at most 2 over all its gates without loss of generality. We invoke \mathcal{A} with the subset being $\{b\}$ and the multiplicative ratio being any constant $c > 1$. If C has no satisfying input, then $b \sim \text{Bernoulli}(0)$. If C has a satisfying input, $b \sim \text{Bernoulli}(p)$ for some $p \geq 1/2^n$. Therefore, a c -factor approximation of the bias of b would decide CIRCEVAL. \square

Acknowledgement A.B. was partially supported by an NRF Fellowship for AI (R-252-000-B13-281), a startup grant (R-252-000-A33-133), and an Amazon Research Award (R-252-000-A61-720). S.G. was supported by A.B.’s NRF Fellowship for AI (R-252-000-B13-281). S.K. was supported by NSF 1846300 CAREER and NSF 1815893 grants. N.V.V. was supported in part by National Science Foundation HDR:TRIPODS-1934884.

References

- [1] J. Acharya, A. Bhattacharyya, C. Daskalakis, and S. Kandasamy. Learning and testing causal models with interventions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9469–9481, 2018.
- [2] A. Bhattacharyya, S. Gayen, S. Kandasamy, A. Maran, and V. N. Variyam. Learning and sampling of atomic interventions from observations. In *International Conference on Machine Learning*, pages 842–853. PMLR, 2020.
- [3] A. Bhattacharyya, S. Gayen, K. S. Meel, and N. Vinodchandran. Efficient distance approximation for structured high-dimensional distributions via learning. *Advances in Neural Information Processing Systems*, 33, 2020.
- [4] A. Bhattacharyya, S. Gayen, E. Price, and N. V. Vinodchandran. Near-optimal learning of tree-structured distributions by chow-liu. *CoRR*, abs/2011.04144, 2020. URL <https://arxiv.org/abs/2011.04144>. ACM STOC 2021.

- [5] C. L. Canonne, I. Diakonikolas, D. M. Kane, and A. Stewart. Testing bayesian networks. *IEEE Trans. Inf. Theory*, 66(5):3132–3170, 2020.
- [6] S. Dasgupta. The sample complexity of learning fixed-structure bayesian networks. *Mach. Learn.*, 29(2-3):165–180, 1997.
- [7] L. Devroye and L. Györfi. *Nonparametric density estimation*. Wiley series in probability and mathematical statistics. Wiley, New York, 1985. ISBN 0471816469.
- [8] Y. Huang and M. Valtorta. Identifiability in causal bayesian networks: A sound and complete algorithm. In *AAAI*, pages 1149–1154, 2006.
- [9] Y. Jung, J. Tian, and E. Bareinboim. Estimating causal effects using weighting-based estimators. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 10186–10193. AAAI Press, 2020.
- [10] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 273–282, 1994.
- [11] L. Malka. How to achieve perfect simulation and a complete problem for non-interactive perfect zero-knowledge. *Journal of Cryptology*, 28(3):533–550, 2015.
- [12] J. Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.
- [13] J. Pearl. *Causality*. Cambridge university press, 2009.
- [14] A. Sahai and S. P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2): 196–249, 2003. doi: 10.1145/636865.636868. URL <https://doi.org/10.1145/636865.636868>.
- [15] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [16] I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1219–1226. AAAI Press, 2006. URL <http://www.aaai.org/Library/AAAI/2006/aaai06-191.php>.
- [17] B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.
- [18] J. Tian. *Studies in causal reasoning and learning*. University of California, Los Angeles, 2002.
- [19] J. Tian and J. Pearl. A general identification condition for causal effects. In *Aaai/iaai*, pages 567–573, 2002.
- [20] T. Watson. The complexity of estimating min-entropy. *computational complexity*, 25(1):153–175, 2016.