Efficient Bayesian network structure learning via local Markov boundary search

Anonymous Author(s)
Affiliation
Address
email

Abstract

We analyze the complexity of learning directed acyclic graphical models from observational data in general settings without specific distributional assumptions. Our approach is information-theoretic and uses a local Markov boundary search procedure in order to recursively construct ancestral sets in the underlying graphical model. Perhaps surprisingly, we show that for certain graph ensembles, a simple forward greedy search algorithm (i.e. without a backward pruning phase) suffices to learn the Markov boundary of each node. This substantially improves the sample complexity, which we show is at most polynomial in the number of nodes. This is then applied to learn the entire graph under a novel identifiability condition that generalizes existing conditions from the literature. We further illustrate the performance of the algorithm, which is easy to implement, in a simulation study. Our approach is general, works for discrete or continuous distributions without distributional assumptions, and as such sheds light on the minimal assumptions required to efficiently learn the structure of directed graphical models from data.

1 Introduction

Learning the structure of a distribution in the form of a graphical model is a classical problem in statistical machine learning, whose roots date back to early problems in structural equations and covariance selection [11, 43, 44]. Graphical models such as Markov networks and Bayesian networks impose structure in the form of an undirected graph (UG, in the case of Markov networks) or a directed acyclic graph (DAG, in the case of Bayesian networks). This structure is useful for variety of tasks ranging from querying and sampling to inference of conditional independence and causal relationships, depending on the type of graph used. In practice, of course, this structure is rarely known and we must rely on *structure learning* to first infer the graphical structure. The most basic version of this problems asks, given n samples from some distribution P that is represented by a graphical model $G = (V, E)$, whether or not it is possible to reconstruct G.

In this paper, we study the structure learning problem for Bayesian networks (BNs). Our main contribution is a fine-grained analysis of a polynomial time and sample complexity algorithm for learning the structure of BNs with potentially unbounded maximum in-degree and without faithfulness. In particular, in our analysis we attempt to expose the underlying probabilistic assumptions that are important for these algorithms to work, drawing connections with existing work on local search algorithms and the conditional independence properties of P.

1.1 Contributions

One of the goals of the current work is to better understand the minimal assumptions needed to identify and learn the structure of a DAG from data. Although this is a well-studied problem, existing

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

theoretical work (see Section 1.2) relies on assumptions that, as we show, are not really necessary. In particular, our results emphasize generic probabilistic structure (conditional independence, Markov boundaries, positivity, etc.) as opposed to parametric or algebraic structure (linearity, additivity, etc.), and hence provide a more concrete understanding of the subtle necessary conditions for the success of this approach.

With this goal in mind, we study two fundamental aspects of the structure learning problem: Identifiability and Markov boundary search. On the one hand, we provide a weaker condition for identifiability compared to previous work, and on the other, we exhibit families of DAGs for which forward greedy search suffices to provably recover the parental sets of each node. More specifically, our contributions can be divided into three parts:

1. *Identifiability* (Theorem 3.1). We prove a new identifiability result on DAG learning. Roughly speaking, this condition requires that the entropy conditioned on an ancestral set $H(X_k \mid A)$ of each node in G is dominated by one of its ancestors. An appealing feature of this assumption is that it applies to general distributions without parametric or structural assumptions, and generalizes existing ones based on second moments to a condition on the local entropies in the model. We also discuss in depth various relaxations of this and other conditions (Appendix B).
2. *Local Markov boundary search* (Algorithm 2, Proposition 4.1). We prove finite-sample guarantees for a Markov boundary learning algorithm that is closely related to the incremental association Markov blanket (IAMB) algorithm, proposed in Tsamardinos et al. [40]. These results also shed light on the assumptions needed to successfully learn Markov boundaries in general settings; in particular, we do not require faithfulness, which is often assumed.
3. *Structure learning* (Algorithm 1, Theorem 5.1). We propose an algorithm which runs in $O(d^3 r \log d)$ time and $O(d^2 r \log^3 d)$ sample complexity, to learn an identifiable DAG G from samples. Here, d is the dimension and $r \leq d$ is the depth of the DAG G, defined in Section 2.

To the best of our knowledge, despite a long history of related work on Markov blanket learning algorithms [e.g. 1, 32, 38], there has been limited theoretical work on the finite-sample properties of IAMB and related algorithms. As an additional contribution of independent interest, we apply our results to the problem of learning polytrees [9, 35].

1.2 Related work

Early approaches to structure learning assumed faithfulness (for the definition, see Appendix C; this concept is not needed in the sequel), which allows one to learn the Markov equivalence class of P [6, 14, 21, 28, 37]. Under the same assumption and assuming additional access to a black-box query oracle, Barik and Honorio [2] develop an algorithm for learning discrete BNs. Barik and Honorio [3] develop an algorithm for learning the undirected skeleton of G without assuming faithfulness. On the theoretical side, the asymptotic sample complexity of learning a faithful BN has also been studied [13, 47]. Brenner and Sontag [4] propose the SparsityBoost score and prove a polynomial sample complexity result, although the associated algorithm relies on solving a difficult integer linear program. Chickering and Meek [7] study structure learning without faithfulness, although this paper does not establish finite-sample guarantees.

An alternative line of work, more in spirit with the current work, shows that G itself can be identified without assuming faithfulness [22, 34, 36, 46], although the associated algorithms lack finite-sample guarantees. Recently, Ghoshal and Honorio [18] translated the equal variance property of Peters and Bühlmann [34] into a polynomial-time algorithm for linear Gaussian models with polynomial sample complexity. Around the same time, Park and Raskutti [31] developed an efficient algorithm for learning a special family of distributions with *quadratic variance functions*. To the best of our knowledge, these algorithms were the first provably polynomial-time algorithm for learning DAGs that did not assume faithfulness. See Ordyniak and Szeider [30] for an excellent discussion of the complexity of BN learning. The algorithm of Ghoshal and Honorio [18] has since been extended in many ways [5, 16, 19, 42].

Our approach is quite distinct from these approaches, although as we show, our identifiability result subsumes and generalizes existing work on equal variances. Additionally, we replace second-moment

assumptions with entropic assumptions, which are weaker and have convenient interpretations in terms of conditional independence, which is natural in the setting of graphical models. Since many of the analytical tools for analyzing moments and linear models are lost in the transition to discrete models, our work relies on fundamentally different (namely, information-theoretic) tools in the analysis. Our approach also has the advantage of highlighting the important role of several fundamental assumptions that are somewhat obscured by the linear case. Finally, we note that although information-theoretic ideas have long been used to study graphical models [e.g. 23–26], these works do not propose efficient algorithms with finite-sample guarantees, which is the main focus of our work.

2 Preliminaries

Notation We use $[d] = \{1, \ldots, d\}$ to denote an index set. As is standard in the literature on graphical models, we identify the vertex set of a graph $G = (V, E)$ with a random vector $X = (X_1, \ldots, X_d)$, and in the sequel we will frequently abuse notation by identifying $V = X = [d]$. For any subset $S \subset V$, $G[S]$ is the subgraph defined by S. Given a DAG $G = (V, E)$ and a node $X_k \in V$, $\mathrm{pa}(k) = \{X_j : (j, k) \in E\}$ is the set of parents, $\mathrm{de}(k)$ is the set of descendants, $\mathrm{nd}(k) := V \setminus \mathrm{de}(k)$ is the set of nondescendants, and $\mathrm{an}(k)$ is the set of ancestors. Analogous notions are defined for subsets of nodes in the obvious way. A source node is any node X_k such that $\mathrm{an}(k) = \emptyset$ and a sink node is any node X_k such that $\mathrm{de}(k) = \emptyset$. Every DAG admits a unique decomposition into r layers, defined recursively as follows: L_1 is the set of all source nodes of G and L_j is the set of all sources nodes of $G[V \setminus \cup_{t=0}^{j-1} L_t]$. By convention, we let $L_0 = \emptyset$ and layer width $d_j = |L_j|$. An ancestral set is any subset $A \subset V$ such that $\mathrm{an}(A) \subset A$. The layers of G define canonical ancestral sets by $A_j = \cup_{t=0}^{j} L_t$. Finally, the Markov boundary of a node X_k relative to a subset $S \subset V$ is the smallest subset $m \subset S$ such that $X_k \perp\!\!\!\perp (S \setminus m) \,|\, m$, and is denoted by $\mathrm{MB}(X_k; S)$ or m_{Sk} for short.

The entropy of a discrete random variable Z is given by $H(Z) = -\sum_z P(Z = z) \log P(Z = z)$, the conditional entropy given Y is $H(Z \,|\, Y) = -\sum_{z,y} P(Z = z, Y = y) \log P(Z = z \,|\, Y = y)$, and the mutual information between Z and Y is $I(Z; Y) = H(Z) - H(Z \,|\, Y)$. For more background on information theory, see Cover and Thomas [8].

Graphical models Let $X = (X_1, \ldots, X_d)$ be a random vector with distribution P. In the sequel, for simplicity, we assume that $X \in \{0, 1\}^d$, however, extensions to categorical random variables and/or continuous random variables is straightforward; see Appendix B.6 for details. We also assume that P is strictly positive, i.e. $P(X = x) > 0$ for all $x \in \{0, 1\}^d$; although again this is not really necessary (see Appendix B.1).

A DAG $G = (V, E)$ is a *Bayesian network* (BN) for P if P factorizes according to G, i.e.

$$P(X) = \prod_{k=1}^{d} P(X_k \mid \mathrm{pa}(k)). \tag{1}$$

Obviously, by the chain rule of probability, a BN is not necessarily unique—any permutation of the variables can be used to construct a valid BN according to (1). A *minimal I-map* of P is any BN such that removing any edge would violate (1). The importance of the factorization (1) is that it implies that separation in G implies conditional independence in P, and a minimal I-map encodes as many such independences as possible (although not necessarily all independences). For more details, a review of relevant graphical modeling concepts is included, see Appendix C.

The purpose of structure learning is twofold: 1) To identify a unique BN G which can be identified from P, and 2) To devise algorithms to learn G from data. This is our main goal.

3 Identifiability of minimal I-maps

In this section we introduce our main assumption regarding identifiability of minimal I-maps of P.

3.1 Conditions for identifiability

For $X_k \in V \setminus A_{j+1}$, define $\mathrm{an}_j(k) := \mathrm{an}(X_k) \cap L_{j+1}$, i.e $\mathrm{an}_j(k)$ denotes the ancestors of X_k in L_{j+1}. By convention, we let $A_0 = \emptyset$. Finally, with some abuse of notation, let $L(X_k) \in [r]$ indicate which layer X_k is in, i.e. $L(X_k) = j$ if and only if $X_k \in L_j$.

Condition 1. For each $X_k \in V$ and for each $j = 0, \cdots, L(X_k) - 2$, there exists $X_i \in \mathrm{an}_j(k)$ such that following two conditions hold:

(C1) $H(X_i \mid A_j) < H(X_k \mid A_j)$,

(C2) $I(X_k; X_i \mid A_j) > 0$.

The idea behind this condition is the following: Suppose we wish to identify the tth layer of G. Condition 1 requires that for every node *after* L_t (represented by X_k), there is at least one node X_i in L_t satisfying (C1) and (C2). The condition (C1) is the operative condition: By contrast, (C2) is just a nondegeneracy condition that says $X_i \not\perp\!\!\!\perp X_k \mid A_j$, which will be violated only when $\mathrm{an}_j(k)$ is conditionally independent of X_k given A_j. One example of such independence between ancestors and descendants is path-cancellation, where effects through different paths are neutralized with delicate parametrization. Not only is this kind of path-cancellation for all of $\mathrm{an}_j(k)$ unlikely, we show in Appendix G that this is essentially the only way (C2) can be violated: If G is a poly-forest, then (C2) always holds, see Lemma G.1.

In Section 3.2, we will discuss this condition in the context of an algorithm and an example, which should help explain its purpose better. Before we interpret this condition further, however, let us point out why this condition is important: It identifies G.

Theorem 3.1. *If there is a minimal I-map G satisfying Condition 1, then G is identifiable from P.*

In order to better understand Condition 1, let us first compare it to existing assumptions such as equal variances [18, 34]. Indeed, there is a natural generalization of the equal variance assumption to Shannon entropy:

Condition 2 (Equal entropy)**.** $H(X_k \mid \mathrm{pa}(k)) \equiv h^*$ is the same for each node $k = 1, \ldots, d$.

Our next result shows that this "equal entropy" condition is a special case of Condition 1:

Lemma 3.2. *Assuming (C2), Condition 2 implies Condition 1.*

In fact, (C1) significantly relaxes Condition 2. The latter implies that all nodes in L_{j+1} have smaller conditional entropy than X_k, whereas (C1) only requires this inequality to hold for at least one ancestor $X_i \in \mathrm{an}_j(k)$. Moreover, something even stronger is true: The equal variance condition can be relaxed to *unequal* variances (see Assumption 1 in [19]), and we can derive a corresponding "unequal entropy" condition. This condition is also a special case of Condition 1. We can also construct explicit examples that satisfy Condition 1, but neither the equal nor unequal entropy condition. For details, see Appendix B.3.

Remark 1. Condition 1 can be relaxed even further: See Appendix B.2 for a discussion along with its corresponding algorithm.

3.2 Algorithmic interpretation

The proof of Theorem 3.1 motivates a natural algorithm to learn the DAG, shown in Algorithm 1. This algorithm exploits the fact that given A_j, nodes within L_{j+1} are mutually independent. Algorithm 1 is a layer-wise DAG learning algorithm. For each layer, it firstly sorts the conditional entropies in ascending order τ, then runs a "Testing & Masking" (T&M) step to distinguish nodes in L_{j+1} from remaining ones (X_k): We use the special node X_i defined in (C1) to detect and *mask* X_k by *testing* conditional independence.

In order to see how Algorithm 1 works, consider the example shown in Figure 1(a). In the first step with $j = 0$, we use marginal entropy (i.e. since $A_0 = \emptyset$) to distinguish L_1 from the remaining nodes. Let $H(X_\ell) := h_\ell$ and assume for simplicity that the nodes are ordered such that $h_1 < h_2 < \cdots < d_6$ (Step 3(b)). Apparently, the inequalities that $h_3 < h_{[4:6]}$ and $h_5 < h_6$ imply Condition 2 does not hold here. Suppose there are no spurious edges, i.e. descendants and ancestors are dependent. Now we can see conditions in Theorem 3.1 are satisfied and the "X_i" for X_2, X_3, X_5 are X_1, X_1, X_4

Algorithm 1 Learning DAG structure

Input: $X = (X_1, \ldots, X_d), \omega$

Output: $\widehat{G}$.

1. Initialize empty graph $\widehat{G} = \emptyset$ and $j = 0$
2. Set $\widehat{L}_j = \emptyset$, let $\widehat{A}_j = \cup_{t=0}^{j} \widehat{L}_t$
3. While $V \setminus \widehat{A}_j \neq \emptyset$:
   (a) For $k \notin \widehat{A}_j$, estimate conditional entropy $H(X_k \mid \widehat{A}_j)$.
   (b) Initialize $\widehat{L}_{j+1} = \emptyset$, $\widehat{S}_{j+1} = \emptyset$. Sort $\widehat{h}_{jk}$ in ascending order and let $\widehat{\tau}^{(0)}$ be the corresponding permutation of $V \setminus \widehat{A}_j$.
   (c) For $\ell \in 0, 1, 2, \ldots$ until $|\widehat{\tau}^{(\ell)}| = 0$: **[T&M step]**
      i. $\widehat{L}_{j+1} = \widehat{L}_{j+1} \cup \{\widehat{\tau}_1^{(\ell)}\}$.
      ii. For $k \notin \widehat{A}_j \cup \widehat{L}_{j+1} \cup \widehat{S}_{j+1}$, estimate $I(X_k; \widehat{\tau}_1^{(\ell)} \mid \widehat{A}_j)$.
      iii. Set $\widehat{S}_{j+1} = \widehat{S}_{j+1} \cup \{k : \widehat{I}_{jk}^{(\ell)} \geq \omega\}$.
      iv. $\widehat{\tau}^{(\ell+1)} = \widehat{\tau}^{(\ell)} \setminus \left(\widehat{L}_{j+1} \cup \widehat{S}_{j+1}\right)$
   (d) For $k \in \widehat{L}_{j+1}$, set $\mathrm{pa}_{\widehat{G}}(k) = \mathrm{MB}(X_k; \widehat{A}_j)$.
   (e) Update $j = j + 1$.
4. Return $\widehat{G}$.

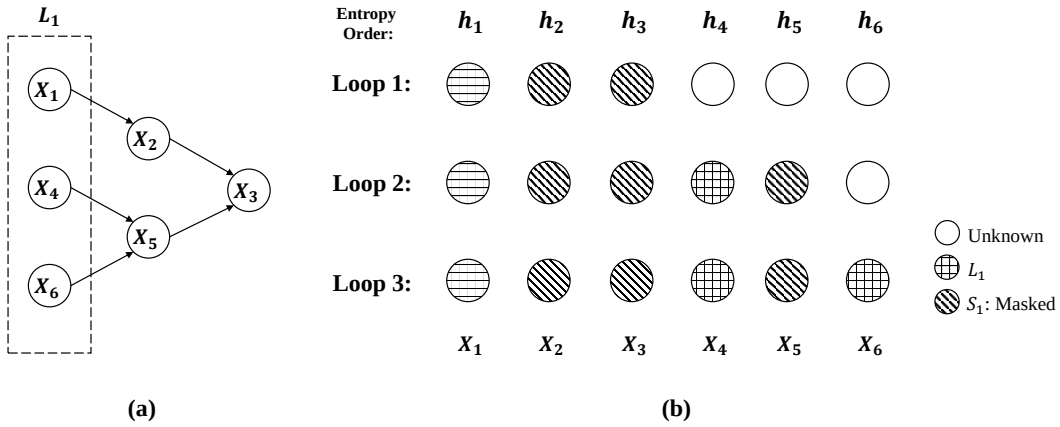

Figure 1: Example for Unequal Entropy algorithm: (a) True DAG with L_1 outlined; (b) Status of the algorithm after each loop, denoted by different patterns of nodes.

respectively. The implementation of Algorithm 1 is visualized in Figure 1(b): In the first loop, X_1 is taken into L_1 and X_2, X_3 are masked due to dependence (Step 3c(iii)). In the second loop, X_4 is added to L_1 and then X_5 is masked. Finally, with X_6 put into L_1, we have identified L_1.

It is worth emphasizing that the increasing order of marginal entropies in this example is purely for simplicity of presentation, and does not imply any information on the causal order of the true DAG. The marginal entropies of nodes need not be monotonic with respect to the topological order of G.

4 Local Markov boundary search

Algorithm 1 assumes that we can learn $\mathrm{MB}(X_k; A_j)$, the Markov boundary of X_k in the ancestral set A_j. This is a well-studied problem in the literature, and a variety of greedy algorithms have been proposed for learning Markov boundaries from data [1, 15, 17, 32, 39, 40], all based on the same

Algorithm 2 Possible Parental Set (PPS) procedure

Input: $X = (X_1, \ldots, X_d)$, k, A, κ

Output: Conditional entropy $\widehat{h}$, Markov boundary $\widehat{m}$

1. Initialize $\widehat{m} = \emptyset$, loop until $\widehat{m}$ does not change:
   (a) For $\ell \in A \setminus \widehat{m}$, estimate conditional mutual information $I(X_\ell; X_k \,|\, \widehat{m})$ by some estimator $\widehat{I}_\ell$.
   (b) Let $\ell^* = \arg\max_{\ell \notin A \setminus \widehat{m}} \widehat{I}_\ell$, if $\widehat{I}_{\ell^*} > \kappa$, set $\widehat{m} = \widehat{m} \cup \{\ell^*\}$.
2. Estimate conditional entropy $H(X_k \,|\, \widehat{m})$ by some estimator $\widehat{h}$.
3. Return conditional entropy estimation $\widehat{h}$ and estimated Markov boundary $\widehat{m}$.

basic idea: Greedily add variables whose association with X_k is the highest. In this section, we establish theoretical guarantees for such a greedy algorithm. In the next section, we use apply this algorithm to reconstruct to full DAG G via Algorithm 1.

To present our Markov boundary search algorithm, we first need to set the stage. Let $X_k \in V$ be any node and A an ancestral set of X_k. We wish to compute $\mathrm{MB}(X_k; A)$ and $H(X_k \,|\, A)$. An algorithm for this is outlined in Algorithm 2. In contrast to many existing local search methods for learning Markov boundaries, this algorithm is guaranteed to return the parents of X_k in G. More specifically, if $X_k \in L_{j+1}$, then $\mathrm{MB}(X_k; A_j) = \mathrm{pa}(X_k)$. For this reason, we refer to Algorithm 2 as *possible parent selection*, or PPS for short. In fact, PPS is exactly the forward phase of the well-known IAMB algorithm for Markov blanket discovery [40] with conditional mutual information used both as an association measure and as a conditional independence test.

Although PPS will always return a valid Markov *blanket*, without a backward phase to remove unnecessary variables added by the forward greedy step, PPS may fail to return a *minimal* Markov blanket, i.e. the Markov boundary. The following condition is enough to ensure no unnecessary variables are included:

Condition 3 (PPS condition)**.** For any proper subset $m \subsetneq \mathrm{MB}(X_k; A)$ and any node $X_\ell \in A \setminus \mathrm{MB}(X_k; A)$, there exists $X_c \in \mathrm{MB}(X_k; A) \setminus m$ such that

$$I(X_k; X_c \,|\, m) > I(X_k; X_\ell \,|\, m).$$

Condition 3 requires that nodes in $\mathrm{MB}(X_k; A)$ always contribute larger conditional mutual information marginally than those that are not in $\mathrm{MB}(X_k; A)$. Thus when we do greedy search to select parents in Algorithm 2, only the nodes in $\mathrm{MB}(X_k; A)$ will be chosen. Therefore, with a proper threshold κ, this set can be perfectly recovered without incorporating any nuisance nodes.

We now present the sample complexity of using PPS to recover Markov boundaries under Condition 3. We will make frequent use of the Markov boundary and its size:

$$m_{Ak} := \mathrm{MB}(X_k; A), \quad M_{Ak} := |m_{Ak}|. \tag{2}$$

In particular, by definition we have $X_k \perp\!\!\!\perp (A \setminus m_{Ak}) \,|\, m_{Ak}$. Under Condition 3, we further define following quantities:

$$\begin{aligned} \widetilde{\Delta}_{Ak} &:= \min_{m \subsetneq m_{Ak}} \left[\max_{c \in m_{Ak} \setminus m} I(X_k; X_c \,|\, m) - \max_{\ell \in A_j \setminus m_{Ak}} I(X_k; X_\ell \,|\, m) \right] > 0, \\ \xi_{Ak} &:= \min_{m \subsetneq m_{Ak}} \min_{c \in m_{Ak} \setminus m} I(X_k; X_c \,|\, m)/2 > 0. \end{aligned} \tag{3}$$

$\widetilde{\Delta}_{Ak}$ is the gap between the mutual information of nodes inside or outside of m_{Ak}. ξ_{Ak} is the minimum mutual information contributed by the nodes in m_{Ak}. The larger these quantities, the easier the $\mathrm{MB}(X_k; A)$ is to be recovered.

Proposition 4.1. *Fix $k \in V$ and let A be any set of ancestors of X_k. Suppose Condition 3 holds. Algorithm 2 is applied for to estimate m_{Ak} and $H(X_k \,|\, A)$ with $\kappa \leq \xi_{Ak}$, we have with $t \leq \min\left(\kappa, \tilde{\Delta}_{Ak}/2\right)$*

$$P\left(\widehat{m}_{Ak} = m_{Ak}, \left|\widehat{H}(X_k \,|\, A) - H(X_k \,|\, A)\right| < t\right) \geq 1 - (M_{Ak} + 2)|A| \frac{\delta_{M_{Ak}}^2}{t^2}.$$

where $\delta^2_{M_{Ak}}$ is the estimation error of conditional entropy defined by (9) *in Appendix E.1, which depends on M_{Ak}.*

A naïve analysis of this algorithm would have a sample complexity that depends on the size of the ancestral set A; note that our more fine-grained analysis depends instead on the size of the Markov boundary $\mathrm{MB}(X_k; A)$. The proof of Proposition 4.1 is deferred to Appendix E.2.

Condition 3 ensures the success of the greedy PPS algorithm. Although this assumption is not strictly necessary for structure learning (see Appendix B.5 for details), it significantly improves the sample complexity of the structure learning algorithm (Algorithm 1). Thus, it is worthwhile to ask when Condition 3 holds: We will take up this question again in Section 5.3.

5 Learning DAGs

Thus far, we have accomplished two important subtasks for learning a DAG: In Section 3, we identified its layer decomposition $L = (L_1, \ldots, L_r)$. In Section 4, we showed that the PPS procedure successfully recovers (ancestral) Markov boundaries. Combining these steps, we obtain a complete algorithm for learning G. In this section, we study the computational and sample complexity of this algorithm; proofs are deferred to the appendices.

We adopt the notations for Markov boundaries as in (2):

$$m_{jk} := \mathrm{MB}(X_k; A_j) \quad M_{jk} := |m_{jk}| \tag{4}$$

Therefore, $X_k \perp\!\!\!\perp (A_j \setminus m_{jk}) \,|\, m_{jk}$. A critical quantity in the sequel will be the size of the largest Markov boundary m_{jk}, which we denote by M:

$$M := \max_{jk} M_{jk} = \max_{jk} |m_{jk}|. \tag{5}$$

This quantity depends on the number of nodes d and the structure of the DAG. For example, if the maximum in-degree of G is 1, then $M = 1$. A related quantity that appears in existing work is the size of the largest Markov boundary *relative to all of* X, which may be substantially larger than M. The former quantity includes both ancestors *and* descendants, whereas m_{jk} only contains ancestors. Analogously, let $M_j = \max_k M_{jk}$.

5.1 Algorithm

By combining Algorithms 1 and 2, we obtain a complete algorithm for learning the DAG G. It consists of two parts:

1. Learning the layer decomposition $(L_1, \ldots, L_r)$ by minimizing conditional entropy and T&M step (Algorithm 1);
2. Learning the parental sets and reducing the size of conditioning sets by learning the Markov boundary $m_{jk} = \mathrm{MB}(X_k; A_j)$ (Algorithm 2).

Specifically, we use PPS (Algorithm 2) to estimate the conditional entropies (Step 3(a)), conditional mutual information (Step 3c(ii)), and the Markov boundary (Step 3(d)). For completeness, the complete procedure is detailed in Algorithm 3 in the supplement. More generally, Algorithm 2 can be replaced with any any Markov boundary recovery algorithm or conditional entropy estimator; this highlights the utility of Algorithm 1 as a separate meta-algorithm for DAG learning.

One missing piece is the choice of estimators for conditional entropy and mutual information in steps (1a) and (2) of the PPS procedure. We adopt the minimax entropy estimator from Wu and Yang [45] by treating (without loss of generality) the joint entropy as the entropy of a multivariate discrete variable, although other estimators can be used without changing the analysis. The complexity of this estimator is exponential in M (i.e. since there are up to 2^M states to sum over in any Markov boundary m_{jk}), so the computational complexity of Algorithm 2 is $O(Md2^M)$. In addition, for the T&M step, there are at most $\max_j d_j$ nodes in each layer to estimate conditional mutual information with remaining at most d nodes, thus this step has computational complexity $O(d \max_j d_j 2^M)$. Thus assuming $M \lesssim \log d$, the overall computational complexity of Algorithm 1 is at most $O\big(r \times (d \times Md2^M + d \times \max_j d_j 2^M)\big) = O(d^3 r \log d)$. Specifically, for r layers, we must estimate the conditional entropy of at most d nodes, and call T&M step once.

5.2 Main statistical guarantees

In order to analyze the sample complexity of Algorithm 3 under Conditions 1 and 3, we introduce following positive quantities:

$$\Delta := \min_j \min_{k \in V \setminus A_{j+1}} (H(X_k \mid A_j) - H(X_i \mid A_j) > 0$$
$$\eta := \min_j \min_{k \in V \setminus A_{j+1}} I(X_k; X_i \mid A_j) > 0$$

where X_i is defined in Condition 1. These two quantities are corresponding to the two conditions (C1) and (C2), which are used to distinguish each layer with its descendants. We also adopt the quantities defined in (3) by setting $A = A_j$ and drop the notation A such that $\widetilde{\Delta}_{jk} := \widetilde{\Delta}_{A_j k}$ and $\xi_{jk} := \xi_{A_j k}$.

Finally we are ready to state the main theorem about sample complexity of Algorithm 1:

Theorem 5.1. *Suppose P satisfies Conditions 1 and 3, and let G be the minimal I-map identified by Theorem 3.1. Let $\widehat{G}$ be output of Algorithm 3 applied with $\omega \leq \eta/2$ and $\kappa \leq \min_{jk} \xi_{jk}$. Denote $\Delta^*_{\omega,\kappa} = \min_{jk}(\Delta/2, \omega, \kappa, \widetilde{\Delta}_{jk}/2)$. If $M \lesssim \log d$ and*

$$n \gtrsim \frac{d^2 r \log^3 d}{(\Delta^*_{\omega,\kappa})^2 \epsilon},$$

then $\widehat{G} = G$ with probability $1 - \epsilon$.

Up to log factors, the sample complexity scales quadratically with the number of nodes d^2 and linearly in the depth $r \leq d$. In the worst case, this is cubic in the dimension. For example, if G is a Markov chain then $M = 1$ and $r = d$, thus it suffices to have $n = \Omega(d^3 \log^3 d)$. In fact, as the proof indicates, the sample complexity depends exponentially on M. A detailed analysis of the dependency on M can be found in the proofs, which are deferred to Appendix E.

Finally, in practice the quantities η, ξ_{jk} needed in Theorem 5.1 may not be known, which hinders the choice of tuning parameters ω, κ. In Appendix E.4 (see Theorem E.2) we discuss the selection of these tuning parameters in a data-dependent way.

5.3 Application to learning polytrees

Learning polytrees is one of the simplest DAG learning problems. This problem was introduced in Rebane and Pearl [35] and in a seminal work, Dasgupta [9] showed that learning polytrees is NP-hard in general. In this section, we note that when the underlying DAG is a polytree (or more generally, a polyforest), Condition 3 is always satisfied, and therefore we have an efficient algorithm for learning identifiable polyforests that satisfy Condition 1.

Recall that a polyforest is a DAG whose skeleton (i.e. underlying undirected graph) has no cycles.

Theorem 5.2. *If P satisfies* (1) *for some polyforest G, then Condition 3 is satisfied.*

The detailed proof can be found in Appendix F.

The crucial property of a poly-forest used in proving Theorem 5.2 is that there exists at most one directed path between any two nodes in the graph. However, the existence of multiple directed paths between two nodes does not necessarily imply that Condition 3 will fail: There are concrete examples of graphs satisfying Condition 3 with arbitrarily many paths between two nodes. An example is given by the DAG $G = (V, E)$ with $V = (Z, X_1, \ldots, X_k, Y)$ such that $Z \to X_i \to Y$ for each i. Thus, this assumption holds more broadly than suggested by Theorem 5.2. It is an interesting problem for future work to study this further.

6 Experiments

We conduct a brief simulation study to demonstrate the performance of Algorithm 1 and compare against some common baselines: PC [37], GES [6]. We focus on the fully discrete setting. All implementation details can be found in Appendix I. We stress that the purpose of these experiments

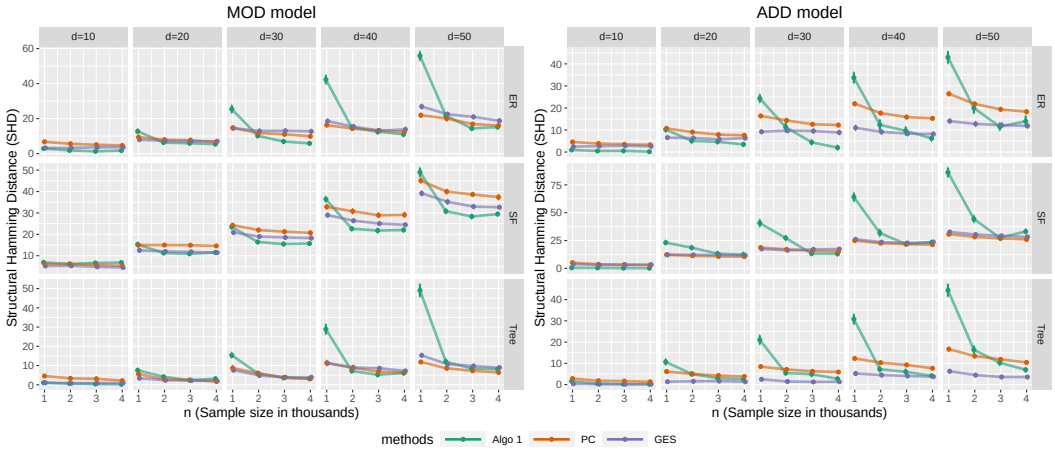

Figure 2: SHD vs sample size n for different dimensions and graph types. Left panel is for MOD model; Right panel is for ADD model.

is simply to illustrate that the proposed algorithm can be implemented in practice, and successfully recovers the edges in G as predicted by our theoretical results.

We evaluate the performance of aforementioned algorithms by Structural Hamming distance (SHD): This is a standard metric for DAG learning that counts the total number of edge additions, deletions, and reversals needed to convert the estimated graph into the reference one. Since PC and GES both return a CPDAG that may contain undirected edges, we evaluate these algorithms favourably by assuming correct orientation for undirected edges wherever they are present.

We simulate DAGs from three graph types: Poly-trees (Tree) , Erdös-Rényi (ER), and Scale-Free (SF) graphs. As discussed in Section 5.3, poly-tree models are guaranteed to satisfy Condition 3, whereas general DAGs such as ER or SF graphs are not, so this provides a test case for when this condition may fail. We generate data according to two models satisfying the "equal entropy" Condition 2. As discussed in Appendix B.3, Condition 2 implies our main identifiability Condition 1.

- "Mod" model (MOD): $X_k = (S_k \mod 2)^{Z_k} \times (1 - (S_k \mod 2))^{1-Z_k}$ where $S_k = \sum_{\ell \in \mathrm{pa}(k)} X_\ell$ with $Z_k \sim \mathrm{Ber}(0.2)$
- Additive model (ADD): $X_k = \sum_{\ell \in \mathrm{pa}(k)} X_\ell + Z_k$ with $Z_k \sim \mathrm{Ber}(0.2)$

Figure 2 (left, right) illustrates the performance in terms of structural Hamming distance (SHD) between the true graph and the estimated graph. As expected, our algorithm successfully recovers the underlying DAG G and performs comparably to PC and GES, which also perform quite well.

7 Conclusion

The main goal of this paper has been to better understand the underlying assumptions required for DAG models to be estimated from data. To this end, we have provided a new identifiability result along with a learning algorithm, which turns out to generalize the existing ones, and analyzed a greedy, local search algorithm for discovering Markov boundaries. This local search algorithm can be used to provably learn the structure of a minimal I-map in polynomial time and sample complexity as long as the Markov boundaries are not too large. Nonetheless, there are many interesting directions for future work. Perhaps the most obvious is relaxing the logarithmic dependence on d in M. It would also interesting to investigate lower bounds on the sample complexity of this model, as well as additional identifiability conditions.

References

[1] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010.

[2] A. Barik and J. Honorio. Learning bayesian networks with low rank conditional probability tables. In *Advances in Neural Information Processing Systems*, pages 8964–8973, 2019.

[3] A. Barik and J. Honorio. Provable efficient skeleton learning of encodable discrete bayes nets in poly-time and sample complexity. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2486–2491. IEEE, 2020.

[4] E. Brenner and D. Sontag. Sparsityboost: A new scoring function for learning bayesian network structure. In *29th Conference on Uncertainty in Artificial Intelligence, UAI 2013*, pages 112–121, 2013.

[5] W. Chen, M. Drton, and Y. S. Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4):973–980, 09 2019. ISSN 0006-3444. doi: 10.1093/biomet/asz049. URL `https://doi.org/10.1093/biomet/asz049`.

[6] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[7] D. M. Chickering and C. Meek. Finding optimal Bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 94–102. Morgan Kaufmann Publishers Inc., 2002.

[8] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[9] S. Dasgupta. Learning polytrees. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 134–141, 1999.

[10] A. P. Dawid. Conditional independence for statistical operations. *Annals of Statistics*, pages 598–617, 1980.

[11] A. P. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[12] M. Drton, B. Sturmfels, and S. Sullivant. *Lectures on algebraic statistics*. Springer, 2009.

[13] N. Friedman and Z. Yakhini. On the sample complexity of learning bayesian networks. In *Uncertainty in Artifical Intelligence (UAI)*, 02 1996. URL `https://arxiv.org/abs/1302.3579`.

[14] N. Friedman, I. Nachman, and D. Peér. Learning Bayesian network structure from massive datasets: The Sparse Candidate algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc., 1999.

[15] S. Fu and M. C. Desmarais. Fast markov blanket discovery algorithm via local learning within single pass. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 96–107. Springer, 2008.

[16] M. Gao, Y. Ding, and B. Aragam. A polynomial-time algorithm for learning nonparametric causal graphs. In *Advances in Neural Information Processing Systems*, 2020.

[17] T. Gao and Q. Ji. Efficient markov blanket discovery and its application. *IEEE transactions on Cybernetics*, 47(5):1169–1179, 2016.

[18] A. Ghoshal and J. Honorio. Learning identifiable gaussian bayesian networks in polynomial time and sample complexity. In *Advances in Neural Information Processing Systems 30*, pages 6457–6466. 2017.

[19] A. Ghoshal and J. Honorio. Learning linear structural equation models in polynomial time and sample complexity. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1466–1475, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[20] Y. Han, J. Jiao, T. Weissman, Y. Wu, et al. Optimal rates of entropy estimation over lipschitz balls. *Annals of Statistics*, 48(6):3228–3250, 2020.

[21] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[22] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696, 2009.

[23] D. Janzing and B. Schölkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

[24] D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

[25] M. Kocaoglu, A. Dimakis, S. Vishwanath, and B. Hassibi. Entropic causal inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[26] M. Kocaoglu, A. G. Dimakis, S. Vishwanath, and B. Hassibi. Entropic causality and greedy minimum entropy coupling. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1465–1469. IEEE, 2017.

[27] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[28] W. Lam and F. Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational intelligence*, 10(3):269–293, 1994.

[29] S. L. Lauritzen. *Graphical models*. Oxford University Press, 1996.

[30] S. Ordyniak and S. Szeider. Parameterized complexity results for exact bayesian network structure learning. *Journal of Artificial Intelligence Research*, 46:263–302, 2013.

[31] G. Park and G. Raskutti. Learning quadratic variance function (QVF) dag models via overdispersion scoring (ODS). *The Journal of Machine Learning Research*, 18(1):8300–8342, 2017.

[32] J. M. Pena, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

[33] J. Peters. On the intersection property of conditional independence and its application to causal discovery. *Journal of Causal Inference*, 3(1):97–108, 2015.

[34] J. Peters and P. Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2013.

[35] G. Rebane and J. Pearl. The recovery of causal poly-trees from statistical data. *arXiv preprint arXiv:1304.2736*, 2013.

[36] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

[37] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

[38] A. Statnikov, N. I. Lytkin, J. Lemeire, and C. F. Aliferis. Algorithms for discovery of multiple markov boundaries. *Journal of Machine Learning Research*, 14(Feb):499–566, 2013.

[39] I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–678, 2003.

[40] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov. Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, volume 2, pages 376–380, 2003.

[41] C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *Annals of Statistics*, 41(2):436–463, 2013.

[42] Y. S. Wang and M. Drton. High-dimensional causal discovery under non-gaussianity. *Biometrika*, 107(1):41–59, 2020.

[43] S. Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557–585, 1921.

[44] S. Wright. The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3): 161–215, 1934.

[45] Y. Wu and P. Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016.

[46] K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 647–655. AUAI Press, 2009.

[47] O. Zuk, S. Margel, and E. Domany. On the number of samples needed to learn the correct structure of a bayesian network. 06 2012. URL `https://arxiv.org/abs/1206.6862`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [No] Since this is a primarily theoretical paper, the immediate negative impacts are expected to be minimal.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In Section 3-5
   (b) Did you include complete proofs of all theoretical results? [Yes] In Appendix D-H
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix I
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Section 6
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Appendix I
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In Appendix I

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]