Interventional idlBNs in DAG-space

Robert Castelo[1][0000−0003−2229−4508]

Department of Medicine and Life Sciences, Universitat Pompeu Fabra, Barcelona, Spain robert.castelo@upf.edu

Abstract. Inclusion-driven structure learning of Bayesian networks, or idlBNs, converges to the generative structure as the sample size grows large and as long as that structure is an acyclic digraph (DAG) over the observed random variables. Because Markov equivalence of Bayesian networks organizes the search space of DAGs in equivalence classes, an obvious choice for such an approach is the greedy equivalence search (GES) algorithm, which carefully traverses the space of essential graphs, the canonical elements of those equivalence classes, following an inclusion path. GES is adapted to data produced by multiple intervention experiments in the greedy interventional equivalence search (GIES) algorithm. The algorithmic complexity of both GES and GIES is in the worst case exponential in the number of vertices, but it can be reduced to polynomial by bounding the vertex degree during the search, albeit at the cost of losing the large-sample optimality guarantee. Inclusion-driven structure learning can also be implemented in the search space of DAGs, as in the hill-climber Monte Carlo (HCMC) algorithm, whose stochastic nature confers the advantage of a polynomial-time bounded algorithmic complexity. Here, we introduce the interventional HCMC (iHCMC) algorithm, an inclusion-driven structure learning algorithm for interventional data in DAG-space. Using synthetic Gaussian data, we verify that iHCMC preserves the large-sample optimality for interventional data with polynomial-time complexity independent of the sparsity of the generative structure.

Keywords: Bayesian network · inclusion-driven structure learning · interventional data.

1 Introduction

Graphical Markov models (GMMs) [44, 28, 15] provide a modular description of a multivariate distribution by means of labeled graphs without loops and multiple edges, whose vertices are in one-to-one correspondence with the random variables of that distribution. Different types of graphs determine different types of GMMs, but they are generally overlapping, i.e., the intersection among those types of models is nonempty. For instance, graphical models determined by chordal graphs are at the intersection between those determined by acylic digraphs (DAGs) and unrestricted undirected graphs, and the intersection of models between chordal graphs and transitive DAGs [2] (those where $a \to b$ and

$b \to c$ imply $a \to c$) is formed by tree conditional independence (TCI) models that are determined by P_4-free chordal graphs [8, 9].

One of the most studied types of GMMs are those determined by DAGs, also known as Bayesian networks, because of their use as a graphical approach to causal inference [45, 33, 34], and their unique and efficient factorization of a joint multivariate distribution in terms of conditional probabilities or densities of each vertex given its parents. When the DAG structure of the Bayesian network of interest is unknown, one can attempt to learn that structure from available data. Because the number of DAGs grows exponentially in the number of vertices [36], structure learning algorithms use heuristic strategies to traverse that search space [13], potentially getting trapped in local maxima. Alternatively, if the number of vertices is small, one can use dynamic programming approaches that efficiently enumerate, score, and find the global optimum [38]. The search for the Bayesian network that best fits the data can also be biased with informative priors on the network structures [7, 31], which can be directly integrated into Markov Chain Monte Carlo (MCMC) procedures when the posterior distribution of Bayesian network structures given the data is of interest [29, 18], as for instance in association rule discovery [4] and web mining [17].

Heuristic approaches can be broadly categorized into constraint-based algorithms that use conditional independence hypothesis tests [39, 23], score-based algorithms that attempt to maximize some sort of goodness-of-fit score [13, 25, 11, 6], and hybrid algorithms that combine the previous two strategies [16, 41]. In addition to its vast size, a feature of the search space of DAGs that makes the learning problem harder is Markov equivalence, where two different DAGs may represent the same set of conditional independence restrictions [10, 1]. One way to approach this complexity is to define the search space in terms of the canonical elements of those equivalence classes, which have been introduced in the literature under different names such as *patterns* [40], completed partially-directed acyclic graphs (*CPDAGs*) [10] and *essential graphs* [1].

Just as we expect a consistent estimator of some quantity of interest to converge in probability to the value that we want to estimate, we may also expect that structure learning algorithms of Bayesian networks have some large-sample optimality guarantees that allow them to converge to the generative structure as the sample size grows large. This is the case of algorithms such as the Peter and Clark (PC) algorithm [39, 23], the greedy equivalence search (GES) [11], or the hill-climber Monte Carlo (HCMC) [25, 6], under the assumption that the generative structure is a DAG over the observed variables. The PC algorithm is constraint-based, while GES and HCMC are score-based.

However, these algorithms preserve this optimality only for independent and identically distributed (iid) observational data. In a causal inference setting, where researchers conduct experiments with *interventions* that set one or more variables to specific values that remove their original causal dependencies, the resulting data points may be independent, but not identically distributed. Learning the structure of Bayesian networks from a mixture of observational and interventional data has been addressed using a simple greedy search algorithm [14] and

adapting GES to this setting in the so-called greedy interventional equivalence search (GIES) algorithm [20].

Due to the way in which graphical manipulations have to be performed to traverse the space of essential graphs, the algorithmic complexity of both GES and GIES is exponential in its worst case [11, 20]. This also happens with the PC algorithm due to the way hypothesis tests are performed [23]. However, if the underlying generative structure is sparse, which is a reasonable assumption in most real-world applications, GES, GIES and PC run in polynomial time. Alternatively, PC, GES and GIES may be run considering a bound on the maximum vertex degree in the graph, which leads to a polynomial-time complexity algorithm, albeit at the cost of losing the large-sample optimality guarantee. On the other hand, HCMC is bounded by polynomial time because of the stochastic nature of its search strategy, independently of the sparsity of the generative structure.

In this paper, we introduce a version of the HCMC algorithm adapted to data from experiments with multiple interventions. Simulating synthetic Gaussian data, we show that just as GIES, this *interventional* HCMC algorithm, which we shall call iHCMC hereafter, preserves its large-sample optimality guarantee with interventional data, while running in polynomial time without the need to restrict the vertex degree in the search space.

The rest of this chapter is organized as follows. In the next section, we introduce background concepts, terminology, and notation on GMMs. In Sections 3 and 4, we briefly review, respectively, inclusion-driven structure learning with the GES and HCMC algorithms, and structure learning from interventional data with the GIES algorithm. In Section 5, we describe the iHCMC algorithm. In Section 6 we empirically show using simulated data the optimal properties of iHCMC. Finally, we conclude this chapter with a discussion in Section 7.

2 Background concepts, terminology and notation

The background concepts, terminology, and notation introduced here have been borrowed from the books of Harary [19], Whitakker [44], Lauritzen [28], and Cox and Wermuth [15]. The reader may consult these books for a more comprehensive account of graphs and GMMs.

2.1 Graphs, paths and separation

A *graph* is a pair $G = (V, E)$ where $V = \{1, \ldots, p\}$ is a finite set of p vertices and $E \subseteq V \times V$ is a subset of edges. Here we consider only graphs that are labeled, where all vertices are distinct, and simple, that is, without loops (edges whose endpoints are the same vertex) and multiple edges (more than one edge between two endpoints). Two vertices $u, v \in V$ are *adjacent* in G if $\{u, v\} \in E$, denoted by $u \sim b$.

A *walk* of length $k \geq 2$ between two vertices x and y in a graph G is a sequence $\pi_{xy} = \langle v_1 = x, \ldots, v_k = y \rangle$ such that $\{v_i, v_{i+1}\} \in E$ for every $i =$

$1, \ldots, k-1$. A *trail* is walk in which all edges $\{v_i, v_{i+1}\} \in E$ are distinct. A *path* is a trail in which all vertices (and therefore also all edges) are distinct. A *circuit* is a nonempty trail whose endpoints are the same vertex. A *cycle* is a circuit in which only the first and last vertices are the same vertex, i.e., $\pi_{xx} = \langle v_1 = x, \ldots, v_k = x \rangle$.

An edge (u, v) is *directed*, also known as an *arc*, if and only if $(u, v) \in E$, but $(v, u) \notin E$. A directed edge between two vertices u and v is represented graphically by an arrow pointing from u toward v, i.e., $u \to v$. A graph $G = (V, E)$ is directed if all edges in E are directed edges. In a directed edge $u \to v$, we call u the *parent* of v and v the *child* of u. All vertices in a directed graph with a common child vertex v are known as the *parent set* of v, denoted by $pa(v)$.

A *directed path* $\pi_{xy} = \langle v_1 = x, \ldots, v_k = y \rangle$ is a direction-preserving path, where every edge on the path points in the same direction, i.e., $v_i \to v_{i+1}$, while a directed cycle is a direction-preserving cycle. An *acyclic directed graph* $G = (V, E)$, popularly known as a DAG, is a directed graph without directed cycles. The *skeleton* of a DAG is the undirected graph obtained by replacing the directed edges by undirected ones, preserving the adjacencies.

We say that a triplet of vertices (x, y, z) in a DAG $G = (V, E)$ forms a *V-configuration* if G has a skeleton such that $x \sim y$ and $y \sim z$, but $x \not\sim z$. We shall distinguish between the following three types of V-configurations: *transition-oriented* where $x \to y \to z$, *source-oriented* where $x \leftarrow y \to z$, and *sink-oriented* where $x \to y \leftarrow z$. The y vertex in a sink-oriented V-configuration is commonly known as a *collider* vertex.

Given three disjoint subsets of vertices $A, B, S \subset V$, where A and B are nonempty, we can classify every path between the vertices in A and B into *active* or *blocked*, according to the membership in S of collider and noncollider vertices of that path. When all paths between vertices in A and B are blocked by S, we say that S *separates* A from B in G and we will denote it by $A \perp_G B | S$.; see [44, 28, 15] for full details on the concept of separation in DAGs, commonly known as *d-separation*. An important intuition behind this and any other definition of separation in graphs is that no matter what subset $S \subset V$ we consider, two vertices $x, y \in V$ in a DAG $G = (V, E)$ directly connected by an edge, i.e., $\{x, y\} \in E$, cannot be separated by S. Therefore, a necessary condition for two vertices x, y to be separated is that they are not adjacent, i.e., $\{x, y\} \notin E$.

2.2 Markov properties, equivalence and inclusion

Let $X \equiv X_V$ be a random vector indexed by a finite set $V = \{1, \ldots, p\}$ with a joint multivariate distribution $P_V \equiv P(X_V)$. Let $G = (V, E)$ be a DAG whose vertices are in one-to-one correspondence with the random variables in X_V. Given three disjoint subsets of random variables $X_A, X_B, X_S \subset X_V$, where X_A and X_B are nonempty, we say that X_A is conditional independent (CI) of X_B given X_S in P_V, and denote it by $X_A \perp\!\!\!\perp X_B | X_S$, or $A \perp\!\!\!\perp B | S$ for short, if and only if the joint density factorizes using the margins defined by A, B and S as, for instance, in $f_{ABS}(a, b, s) = f_{AS}(a, s) f_{BS}(b, s) / f_S(s)$ for all s with $f_S(s) > 0$.

Using d-separation relationships $\perp_G$, we can encode or represent, in a DAG G, a subset of the CI restrictions that hold in P_V. We say that P_V obeys the *global Markov property* relative to a DAG G, or that P_V is *Markov over a DAG* G, if for every triplet of disjoint subsets $A, B, S \subset V$, where A and B are nonempty, $A \perp_G B|S \Rightarrow A \perp\!\!\!\perp B|S$ in P_V. Additional Markov properties exist using graphical criteria other than d-separation; see [28] for further details.

We call the family of all joint multivariate distributions P_V Markov over a DAG G, denoted by $\mathcal{M}(G)$, the GMM determined by G, or to simplify terminology and notation, the Bayesian network G. A multivariate density function $f(x)$ for $P_V \in \mathcal{M}(G)$ also obeys the global Markov property relative to a DAG G, if it admits the following unique factorization in terms of conditional densities of each random variable given its parents in G [28]:

$$f_G(x) = \prod_{i \in V}^{p} f_G(x_i|x_{pa(i)})\,. \tag{1}$$

A distinctive feature of Bayesian networks, with respect to some other types of GMMs such as those determined by undirected graphs, is that two different Bayesian networks G and G' may represent the same set of CI restrictions as in, e.g., $a \leftarrow b \leftarrow c$, $a \leftarrow b \rightarrow c$ and $a \rightarrow b \rightarrow c$, which all represent $X_a \perp\!\!\!\perp X_c|X_b$. In such a case, $\mathcal{M}(G) = \mathcal{M}(G')$, and one says that G and G' are *Markov equivalent*. From a graphical perspective, two different DAGs G and G' are Markov equivalent if and only if they have the same skeleton and the same sink-oriented V-configurations [43].

The canonical element of a Markov equivalence class of DAGs is represented by an essential graph (EG) [1], formed by directed and undirected edges without partially directed cycles. In an EG, an edge is directed if an only if that edge has the same orientation in all DAGs from the equivalence class it represents, otherwise it is undirected. Given a DAG G in an equivalence class with two or more members, G should have at least one arc that can be reversed in such a way that the resulting DAG remains in the same equivalence class. Such arcs are said to be *covered* and are characterized as follows [10].

Definition 1. *(Covered arc) Given a DAG $G = (V, E)$, an arc $a \rightarrow b$ is covered in G if $pa(b) = \{a\} \cup pa(a)$.*

The previous definition means that an arc $a \rightarrow b$ is covered if and only if the parent sets of a and b are identical except for the a vertex, which is a parent of b, but it cannot be a parent of itself. Covered arcs provide the following additional characterization of Markov equivalence.

Lemma 1. *(Lemma 2.1 [6]) Given two different Markov equivalent DAGs G and G', there exists a sequence $L_1, \ldots, L_n$ of DAGs such that $L_1 = G$ and $L_n = G'$ and L_{i+1} is obtained from L_i by reversing a covered edge in L_i, for $i = 1, \ldots, n-1$.*

From the previous definitions of Markov equivalence, it follows that two DAGs G and G' with a different number of arcs cannot be Markov equivalent. Two

extreme such cases are the complete DAG G_c, where all vertices are adjacent and no CI restriction can be represented, and the empty DAG $G_\emptyset$, where no arc is present and, therefore, all possible CI restrictions are represented. Clearly, the family of all P_V Markov over $G_\emptyset$ is more constrained than the one that is Markov over G_c, and therefore we may say that $\mathcal{M}(G_\emptyset)$ is included in $\mathcal{M}(G_c)$, i.e., $\mathcal{M}(G_\emptyset) \subset \mathcal{M}(G_c)$ [26, 25, 6]. We call this relationship a *Markov inclusion order*, and can also be defined using only subsets of CI restrictions as follows. Given a DAG $G = (V, E)$,

$$\mathcal{M}'(G) = \{(A, B, S) : A, B \neq \emptyset \wedge A \perp_G B | S\}, \tag{2}$$

defines the set of CI restrictions that could be read off the DAG G. Under this definition, $\mathcal{M}'(G_c) \subset \mathcal{M}'(G_\emptyset)$. The collection of subsets of CI restrictions for all DAGs on p vertices $\{\mathcal{M}'(G_1), \ldots, \mathcal{M}'(G_k)\}$ forms a partial ordered set, or *poset*, with a partial order relation defined by Markov inclusion [6, Fig. 2]. A sequence of DAGs $G_1, \ldots, G_k$ forms an *inclusion path* if either $\mathcal{M}'(G_1) \subseteq \mathcal{M}'(G_2) \subseteq \ldots \subseteq \mathcal{M}'(G_k)$ or $\mathcal{M}'(G_1) \supseteq \ldots \supseteq \mathcal{M}'(G_{k-1}) \supseteq \mathcal{M}'(G_k)$.

Given two DAGs G and G' that are not Markov equivalent, deciding whether $\mathcal{M}'(G) \subset \mathcal{M}'(G')$, $\mathcal{M}'(G) \supset \mathcal{M}'(G')$ or whether G and G' are not in inclusion, was an open problem for a long time. In 1988, Verma and Pearl [42] attempted to give necessary and sufficient conditions to characterize the inclusion order of Bayesian networks. Later in 1997, Chris Meek in his PhD thesis [30] provided the following conjecture on the inclusion problem.

Conjecture 1. (Meek's conjecture [30]) Given two Bayesian networks G and G', $\mathcal{M}'(G) \subseteq \mathcal{M}'(G')$ if and only if there exists a sequence of DAGs $G_1, \ldots, G_n$ such that $G_1 = G$, $G_n = G'$ and the G_{i+1} is obtained from G_i by either reversing a covered arc or removing an arc, for $i = 1, \ldots, n-1$.

In 2001, Kočka, Bouckaert and Studený [26] showed that the conditions given by Verma and Pearl in [42] were necessary but not sufficient, and gave a proof to Meek's conjecture for the particular case in which G and G' differ in at most one adjacency. Finally, in 2002, Max Chickering [11] provided a general proof of Meek's conjecture by means of the following theorem.

Theorem 1. *(Theorem 4 [11]) Let G and G' be any pair of DAGs such that $\mathcal{M}'(G) \subseteq \mathcal{M}'(G')$. Let r be the number of arcs in G that have opposite orientation in G', and let m be the number of arcs in G' that do not exist in either orientation in G. There exists a sequence of at most $r+2m$ distinct arc reversals and additions in G' with the following properties:*

1. *Each arc reversed is a covered arc.*
2. *After each reversal and addition, G' is a DAG and $\mathcal{M}'(G) \subseteq \mathcal{M}'(G')$.*
3. *After all reversals and additions $G = G'$.*

As we shall see in the next section, the Markov inclusion order is relevant for structure learning of Bayesian networks, because it enables building learning algorithms with a large-sample optimality guarantee.

3 Inclusion-driven structure learning

Score-based algorithms for learning the structure of a Bayesian network from data consist of a scoring metric and a search strategy. Several scoring metrics have been introduced for discrete and continuous data, e.g., [37, 3, 13, 22, 27], based on the unique factorization of the probability mass or density function that one may derive from a given DAG structure (see Eq. 1 in the previous section). A useful property of some of these score metrics, such as BDe [22], BGe [27] or BIC [37], is local consistency. A score metric is *locally consistent* [11, 6] if, given data sampled from a joint multivariate distribution P_V, it increases after adding an arc that removes a CI restriction that does not hold in P_V, and it decreases after adding an arc that removes a CI restriction that holds in P_V, i.e., it will increase when an *unnecessary* arc is removed.

A straightforward search strategy is a greedy hill-climbing algorithm that starts from a DAG G, typically the empty DAG without edges $G_\emptyset$, i.e., $G = G_\emptyset$, scores it, generates all possible neighboring DAGs with one more arc, scores them, and selects the one with the highest score; let us call it G'. If the score for G' is higher than the score for G, then $G = G'$, we start again generating all possible neighboring DAGs and continue in this loop until the score does not improve. When the algorithm stops, G is the DAG with the highest score found by the algorithm.

The set of neighboring DAGs can be generated in different ways, where a simple one is to create all DAGs derived from adding one arc in each empty adjacency, as long as it does not introduce a directed cycle, removing every present arc, and reversing every present arc, as long as it does not introduce a directed cycle. It has been shown [11, 6] that this simple hill-climbing algorithm gets easily stuck in local maxima, even when using a locally consistent scoring metric.

However, if the search strategy is such that the neighboring DAGs generated at each step enable following an inclusion path, then in the limit of the size of the sample, the hill-climbing algorithm will converge to the generative structure [26, 25, 12, 11, 6]. This is the case for the hill-climber Monte carlo (HCMC) algorithm [25, 6] that works in the space of DAGs, and the greedy equivalence search (GES) algorithm that works in the space of EGs [11]. A distinctive feature of the HCMC algorithm is that it is stochastic, which enables following an inclusion path with positive probability, while bounding its algorithmic complexity to polynomial time. On the other hand, GES deterministically attempts to follow an inclusion path, but the graphical operations required to generate neighboring EGs have a worst-case exponential algorithmic complexity.

Structure learning approaches other than the previous greedy strategies can also lead to optimal algorithms, as long as they attempt to follow an inclusion path. This is the case of the constraint-based PC algorithm [39] or the MCMC-based $e\text{MC}^3$ algorithm [25, 6].

4 Interventional structure learning

Because two different Markov equivalent DAGs G and G' also lead to two equivalent factorizations $f_G(x) = f_{G'}(x)$, factorizing a joint multivariate distribution according to a DAG is necessary but not sufficient to make a causal interpretation of that DAG. To approach such a causal interpretation, we may consider *interventional probability distributions* using the *do* operator [35, pg. 70]:

$$f_G(Y = y|do(X = x))\,, \tag{3}$$

by which we obtain the density of $Y = y$ when we intervene in X, by setting the value $X = x$. Given a DAG $G = (V, E)$, we call the *intervention targets* [20] the subset of vertices $I \subseteq V$ indexing the variables $X_I \subseteq X_V$ that we want to intervene in. When we *intervene* in one variable with $do(X = x)$, we will assume that such intervention is *modular* [35, pg. 63], i.e., changing the causal mechanism in one of the variables in the system does not change the causal mechanism in other variables. More intuitively, modularity implies that the intervention only affects the incoming edges of the intervened variable by removing them, while the rest of the graph structure remains intact. This notion can be formalized in the following definition of an *intervention graph* [20, Definition 5].

Definition 2. *(Intervention graph) Let $G = (V, E)$ be a DAG with vertex set V and edge set E, and $I \subseteq V$ a subset of intervention targets. The intervention graph of G is the DAG $G^{(I)} = (V, E^{(I)})$, where $E^{(I)} := \{(a, b)|(a, b) \in E, b \notin I\}$.*

Modularity also implies that if we intervene in a subset of variables $X_I \subseteq X_V$, when $i \notin I$ then $f_G(x_i|x_{pa(i)})$ remains unchanged, while if $i \in I$ then $f_G(x_i|x_{pa(i)}) > 0$ only if $do(X_i = x_i)$ and $f_G(x_i'|x_{pa(i)}) = 0$ for every other $x_i' \neq x_i$. Consequently, altering the graph structure after an intervention leads to the following *truncated factorization* [35, pg. 24][20]:

$$f_G(X_V = x_V|do(X_I = x_I)) = \prod_{i \notin I} f(x_i|x_{pa(i)}) \prod_{i \in I} \tilde{f}(x_i)\,, \tag{4}$$

where $\tilde{f}(x_i)$ denotes the joint product density for $do(X_i = x_i)$. When $I = \emptyset$, then $f_G(X_V = x_v|do(X_I = x_I)) = f_G(X_V)$, including the observational density as a specific case of a truncated factorization without interventions.

A *family of targets* $\mathcal{I} = \{I_j\}_{j=1}^{J}$ [20, pg. 2412] is a collection of subsets of targets. Without loss of generality, assuming that I is either the empty set or a single-vertex intervention target, an example of a family of targets could be $\mathcal{I} = \{\emptyset, \{1\}, \{2\}, \{3\}\}$, for a DAG G with at least 3 vertices.

Given a family of targets $\mathcal{I}$ and a DAG G, an interventional Bayesian network, or simply an interventional DAG, is a family of joint multivariate distributions $\mathcal{M}_{\mathcal{I}}(G)$ Markov over G and therefore $\mathcal{M}_{\emptyset}(G) = \mathcal{M}(G)$. Two different DAGs G and G' are $\mathcal{I}$-Markov equivalent when $\mathcal{M}_{\mathcal{I}}(G) = \mathcal{M}_{\mathcal{I}}(G')$. In graphical terms, this implies that for every $I \in \mathcal{I}$, the two intervention graphs $G^{(I)}$ and $G'^{(I)}$ have

the same skeleton and the same sink-oriented V-configurations [20]. This definition of Markov equivalence for interventional DAGs provides a finer partition of Markov equivalence classes.

An *interventional data set* D of sample size n generated by one or more intervention graphs $G^{(I_1)}, \ldots, G^{(I_k)}$ through a family of targets $\mathcal{I} = \{I_1, \ldots, I_k\}$, is the matrix of values $D = \{x_{ij}\}_{n \times p}$, where $x_{i\cdot}$ are independent but not identically distributed data points sampled from different interventions, specified for every row of that matrix by the *target index vector* $\mathcal{T} = (T^{(1)}, \ldots, T^{(n)})$ with $T^{(i)} = I_l$ and $I_l \in \mathcal{I}$, assuming that each target $I \in \mathcal{I}$ occurs at least once in $\mathcal{T}$. When at least one $T^{(i)} = \emptyset$ and one $T^{(j)} \neq \emptyset$, then D is contains a mixture of observational and interventional data.

Given a data set D of multivariate Gaussian data, resulting from a mixture of observational and interventional data, specified by a family of targets $\mathcal{I}$ and a target index vector $\mathcal{T}$, a Bayesian information criterion (BIC) score for this type of data was introduced in [20] and shown to be *consistent* in [21], i.e., in the limit $n \to \infty$ of the sample size of D, the DAG G maximizing the BIC is the generative structure G from which D was sampled.

These results allowed Hauser and Bühlmann in [20] to derive a generalization of the GES algorithm for the structure learning of Bayesian networks from interventional data, in what they called the Greedy Interventional Equivalence Search (GIES) algorithm.

5 The interventional HCMC algorithm

Here, we adapt HCMC to interventional data. The concept of Markov equivalence for interventional DAGs leads to a new, more restrictive concept of a covered arc, the *interventional covered arc*, or $\mathcal{I}$-covered arc for short.

Definition 3. *($\mathcal{I}$-covered arc) Given a DAG $G = (V, E)$ and a family of targets $\mathcal{I}$, $a \to b$ is $\mathcal{I}$-covered in G if $pa(b) = \{a\} \cup pa(a)$ and $I \cap \{a, b\} = \emptyset$ for all $I \in \mathcal{I}$.*

We can now write the following interventional repeated covered arc reversal (iRCAR) algorithm, whose input is a DAG, a parameter r of the maximum number of $\mathcal{I}$-covered arc reversals, and a family of targets $\mathcal{I}$.

```
begin algorithm ircar(dag, r, targets)
01      rr ← rnd(0, r)
02      for i ← 0 to rr do
03            cov_arcs ← covered_arcs(dag)
04            mask ← cov_arcs in targets
05            i_cov_arcs ← cov_arcs[not mask]
06            j ← rnd(0, length(i_cov_arcs)-1)
07            dag ← reverse_arc(dag, i_cov_arcs[j])
08      endfor
09      return dag
endalgorithm
```

In this algorithm, the function `covered_arcs()` returns a vector of covered arcs from the input DAG, while the function `reverse_arc()` returns the DAG given in the first argument, reversing the arc specified in the second argument.

Because the HCMC algorithm works in DAG-space, we can readily use the BIC score for interventional data described in [20, 21]. Using this BIC score and the previous iRCAR algorithm we can generalize as follows the HCMC algorithm to interventional data, in what we shall call the *interventional HCMC* algorithm, or iHCMC for short.

```
begin algorithm ihcmc(data, r, maxtrials, targets, target_index)
01      dag ← emptydag(data)
02      trials ← 0
03      local_maximum ← false
04      while not local_maximum do
05            dag1 ← ircar(dag, r, targets)
06            ne_dags ← nicr(dag1, targets)
07            dag1 ← argmax(score(data, ne_dags,
08                                targets, target_index))
09            s ← score(data, dag, targets, target_index)
10            s1 ← score(data, dag1, targets, target_index)
11            local_maximum ← s1 <= s
12            if not local_maximum then
13                  dag ← dag1
14                  trials ← 0
15            else if trials < maxtrials then
16                  dag ← ircar(dag, r, targets)
17                  local_maximum ← false
18                  trials ← trials + 1
19            endif
20      endwhile
21      return dag
endalgorithm
```

The input parameters for the iHCMC algorithm are a data set, a parameter r of the maximum of $\mathcal{I}$-covered arc reversals, a maximum number of trials escaping from local maxima, a family of targets $\mathcal{I}$ and a target index vector $\mathcal{T}$. The `nicr()` function, given an input DAG, generates all neighboring DAGs with one arc added, one arc removed, and every non-$\mathcal{I}$-covered arc reversed. The `score()` function implements the BIC score function for interventional data described in Section 4. The `argmax()` function returns the DAG with the highest BIC score.

6 Experimental evaluation

In this section, we show the results of assessing the performance of iHCMC and other competing algorithms in learning the structure of Bayesian networks from mixtures of simulated observational and interventional data. We have replicated

the experimental evaluation strategy used in the assessment of the GIES algorithm [20] with a few modifications that we will specify where they apply.

We have used the R package `pcalg` [24] to simulate Bayesian networks, Gaussian data from them, and run the algorithms we compare with iHCMC, for which we have developed our own implementation in R. As in [20], we use as a baseline comparison a learning method that does not have a large-sample optimality guarantee, the greedy DAG search (GDS) algorithm that operates in DAG-space by simply adding, removing or reversing arcs, and which is implemented in the function `gds()` of the `pcalg` package. From this package, we have also used the functions `ges()` and `gies()` to run, respectively, the GES and GIES algorithms. We have used the BIC score for Gaussian observational and interventional data described in [20] and implemented in the `pcalg` package through the object classes `GaussL0penObsScore` and `GaussL0penIntScore`, respectively.

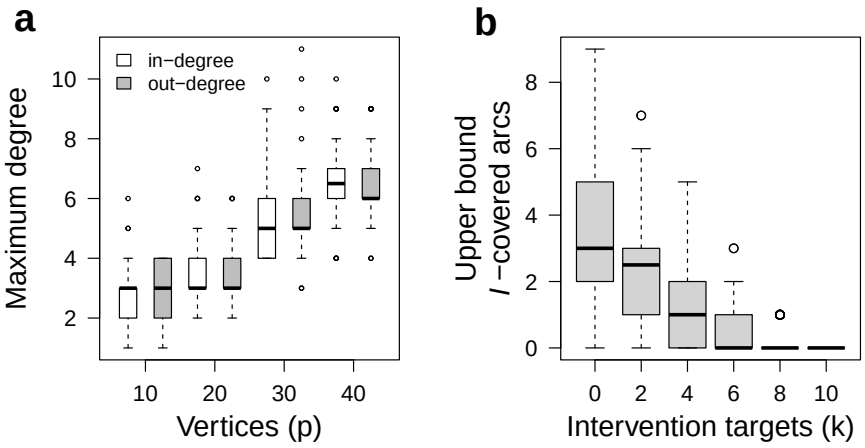

Fig. 1. Simulated intervention DAGs. In (a) maximum in- and out-degree across 100 DAGs for four different combinations of dimension and sparsity rate. In (b) upper bound of the number of $\mathcal{I}$-covered edges as a function of the number k of intervention vertices for 100 DAGs of $p = 10$ vertices.

To simulate random Gaussian Bayesian networks, their parameters and multivariate observational and interventional data from them, we have used methods described in [20] and implemented in the functions `r.gauss.pardag()` and `rmnorm.ivent()` of the `pcalg` package. Unlike the evaluation of the GIES algorithm in [20], where they ran the GDS algorithm with a BIC scoring function only for interventional data, we have also run GDS with a scoring function for observational data. To distinguish between these two regimes of the GDS algorithm, we have labeled the observational data regime with `GDS`, and the interventional one with `iGDS`. We have used the following parameters in our simulations.

- Four different combinations of DAG dimension and sparsity rate $(p, s) \in \{(10, 0.2), (20, 0.1), (30, 0.1), (40, 0.1)\}$ and for each of them we have simulated 100 random DAGs.
- Families of intervention targets of the form $\mathcal{I} = \{\emptyset, I_1, \ldots, I_k\}$, where $I_1, \ldots, I_k$ are k different, randomly chosen intervention targets of size 1 and $k = \{0, 0.2p, 0.4p, 0.6p, 0.8p, p\}$. When $k = 0$, i.e., without intervention targets, the simulated data is purely observational.
- Eight increasing sample sizes $n \in \{50, 100, 200, 500, 1000, 2000, 5000, 10000\}$.

Figure 1b shows the upper bound of the number of $\mathcal{I}$-covered edges as a function of the number k of intervention targets of size 1, across the 100 randomly generated DAGs. It partly reproduces Figure 8 in [20] with 100 instead of 1000 DAGs, because the number of non-$\mathcal{I}$-essential arrows reported in that figure is an upper bound for the number of $\mathcal{I}$-covered edges at any given stage of the iHCMC algorithm.

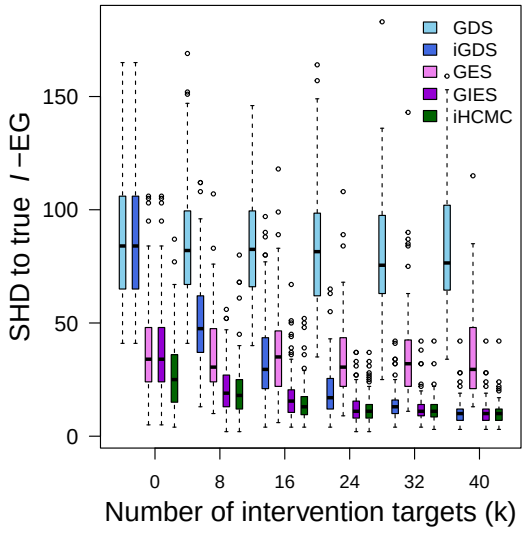

Fig. 2. Structural hamming distance (SHD) between estimated and true $\mathcal{I}$-EGs as function of the number k of single-vertex intervention targets, across 100 random DAGs generated of $p = 40$ vertices. Each data set contains $n = 1000$ data points.

As a measure of divergence between the underlying simulated DAG and the one estimated with one of the learning algorithms, we use the structural Hamming distance (SHD) implemented in the function `shd()` of the `pcalg` package [23]. We ran GDS, iGDS, GES, GIES and iHCMC across the data sets simulated from the different combinations of the parameters given before, and calculated the SHD with respect to the generative $\mathcal{I}$-essential graph ($\mathcal{I}$-EG). In the case of

the GDS, iGDS and iHCMC algorithms, we first transform the resulting DAG into its corresponding $\mathcal{I}$-EG, and then calculate the SHD.

We ran iHCMC with arguments `r=20` and `maxtrials=5` throughout all simulations. Figure 2 shows for $p = 40$ how the SHD decreases for GIES, iGDS, and iHCMC as the number of single-vertex intervention targets increases, while the SHD does not improve for GDS and GES, which are only designed to work with observational data. Similar results are obtained with $p = \{10, 20, 30\}$.

As observed in [20], GES and GIES, and GDS and iGDS in our simulations, perform identically with observational data ($k = 0$), while iGDS, GIES, and iHCMC, perform similarly when the number of single-vertex intervention targets exceeds 50% of the vertices. Likewise, the similar performance of iGDS to GIES and iHCMC is due to the finer partition of interventional Markov equivalence classes, which become singletons when $k = p$ because every DAG becomes its own $\mathcal{I}$-EG. The performance of iGDS underscores the importance of using a scoring function that takes into account the intervention information at each data point.

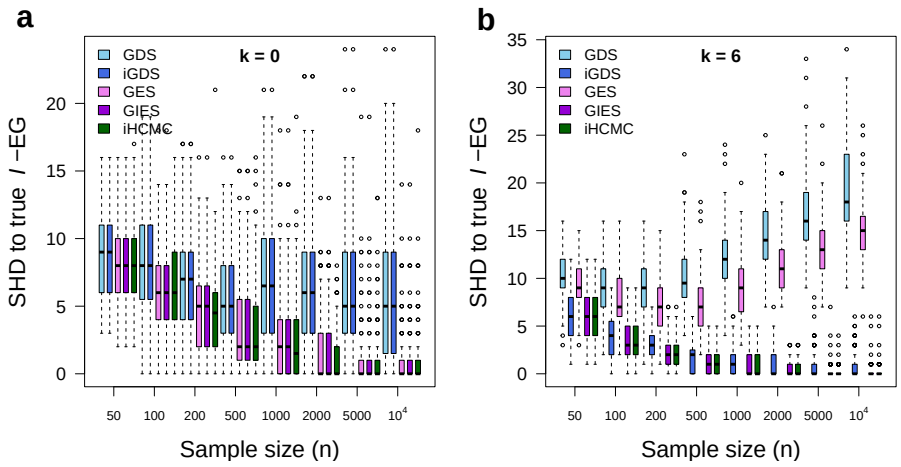

Fig. 3. Structural hamming distance (SHD) between estimated and true $\mathcal{I}$-EGs as function of the sample size, for two different number of intervention targets of size $m = 1$ on DAGs with $p = 10$ vertices, $k = 0$ (a) and $k = 6$ (b).

Figure 3 shows the convergence of the different algorithms to the generative structure, as the sample size grows large, in terms of SHD, for two different numbers of single-vertex intervention targets ($k = \{0, 6\}$), where $k = 0$ corresponds to observational data. We may see that with observational data ($k = 0$), GES, GIES, and iHCMC converge to the generative structure in the limit of the size of the sample, while this does not happen with GDS or iGDS, since these are not inclusion-driven algorithms and therefore have no large-sample optimality guar-

antees. With interventional data, GDS and GES learn structures that diverge as the sample size and the number of intervention targets increases.

We simulated DAGs with four different dimensions and sparsity rates, which may result in different degrees of connectivity among the vertices. Sparsity and, in particular, the underlying vertex degree should be taken into account if we are going to restrict the maximum number of edges per vertex in the learning algorithm. Figure 1a shows the maximum in- and out-degree of the 100 simulated DAGs for each of the four different dimensions. We can see that for these simulated DAGs, the maximum vertex degree increases with the dimension.

We have assessed the influence of restricting the vertex degree in the search space by running the algorithms again on the data simulated from DAGs with $p = 10$ vertices and setting GES and GIES to explore the search space with a maximum vertex degree of 3, which should be sufficient for a majority of the underlying DAGs, as shown in Figure 1a. The results in Figure 4 reveal that GES (for $k = 0$) and GIES lose their optimality observed in the previous simulation in Figure 3.

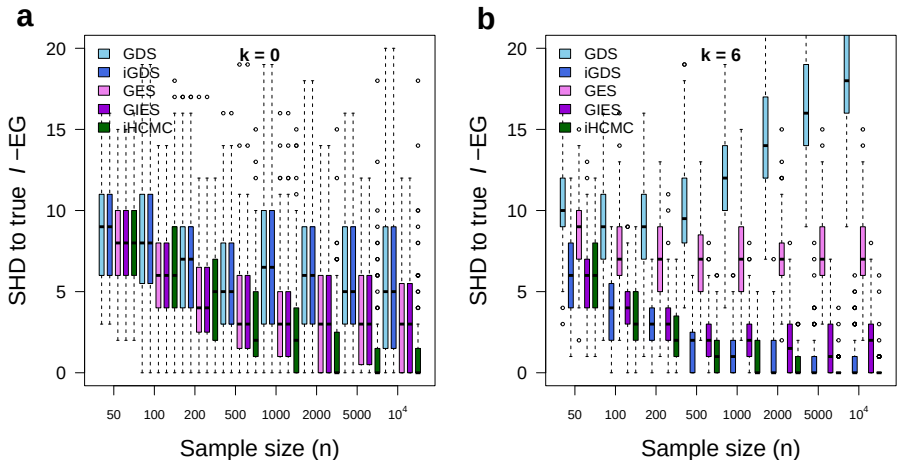

Fig. 4. Structural hamming distance (SHD) between estimated and true $\mathcal{I}$-EGs as function of the sample size, for two different number of intervention targets of size $m = 1$ on DAGs with $p = 10$ vertices, $k = 0$ (a) and $k = 6$ (b). Here, GES and GIES were run with a maximum vertex degree of 3 in the estimated graph.

7 Discussion

Inclusion-driven learned Bayesian networks, or idlBNs[1], have a large-sample optimality guarantee that becomes useful in applications where the available sam-

[1] Pronounced *ideal BNs*.

ple size is high, such as in the prediction of spliceosome binding sites on DNA sequences [5]. The trade-off between greediness and randomness that the HCMC algorithm provides has been shown [32] to be also a useful feature when learning from data with a large number of local optima.

Data from intervention experiments, such as in clinical trials, molecular manipulations, or public policy, convey information and meet assumptions that can be harnessed to perform causal inference. Adapting learning algorithms to interventional data contributes to exploiting the results produced by often nontrivial and expensive experiments.

Here we have adapted HCMC, an inclusion-driven structure learning algorithm of Bayesian networks in DAG-space [25, 6], to interventional data in the iHCMC algorithm, building on the previous work that led to the development of the GIES algorithm, which performs the same task in EG-space.

We have empirically verified using simulated synthetic data with interventions that iHCMC preserves the same optimal properties as the GIES algorithm and has the advantage that it does not require to bound the maximum vertex degree in the search space to keep the algorithmic complexity in polynomial time.

Acknowledgments. This work was supported by the research project PID2019-105595GB-I00 funded by the MICIU/AEI/10.13039/501100011033. The author thanks the anonymous reviewers for useful comments and suggestions that have helped improve this chapter and the developers and contributors of the R package `pcalg`, which has greatly facilitated running the experimental simulations. The author also wishes to express his most sincere gratitude to Arno Siebes for his mentorship, guidance, and support throughout the pre-doctoral training and beyond.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Andersson, S., Madigan, D., Perlman, M.: A characterization of Markov equivalence classes for acyclic digraphs. Annals of Statistics **25**, 505–541 (1997)
2. Andersson, S.A., Madigan, D., Perlman, M.D., Triggs, C.M.: On the relation between conditional independence models determined by finite distributive lattices and by directed acyclic graphs. Journal of Statistical Planning and Inference **48**(1), 25–46 (1995)
3. Buntine, W.: Theory refinement on Bayesian networks. In: D'Ambrosio, B.D., Smets, P., Bonissone, P.P. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 52–60. Morgan Kaufmann (1991)
4. Castelo, R., Feelders, A., Siebes, A.: Mambo: Discovering association rules based on conditional independencies. In: Advances in Intelligent Data Analysis: 4th International Conference, IDA 2001 Cascais, Portugal, September 13–15, 2001 Proceedings 4. pp. 289–298. Springer-Verlag (2001). https://doi.org/https://doi.org/10.1007/3-540-44816-0_29
5. Castelo, R., Guigó, R.: Splice site identification by *idl*BNs. Bioinformatics **20**(Suppl 1), i69–i76 (2004). https://doi.org/10.1093/bioinformatics/bth932

6. Castelo, R., Kočka, T.: On inclusion-driven learning of Bayesian networks. Journal of Machine Learning Research **4**(Sep), 527–574 (2003)
7. Castelo, R., Siebes, A.: Priors on network structures. Biasing the search for Bayesian networks. International Journal of Approximate Reasoning **24**(1), 39–57 (2000). https://doi.org/10.1016/S0888-613X(99)00041-9
8. Castelo, R., Siebes, A.: A characterization of moral transitive acyclic directed graph Markov models as labeled trees. Journal of Statistical Planning and Inference **115**(1), 235–259 (2003). https://doi.org/10.1016/S0378-3758(02)00143-X
9. Castelo, R., Wormald, N.: Enumeration of P_4-free chordal graphs. Graphs and Combinatorics **19**(4), 467–474 (2003). https://doi.org/10.1007/s00373-002-0513-9
10. Chickering, D.M.: A transformational characterization of equivalent Bayesian networks. In: Besnard, P., Hanks, S. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 87–98. Morgan Kaufmann (1995)
11. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research **3**(Nov), 507–554 (2002)
12. Chickering, D.M., Meek, C.: Finding optimal Bayesian networks. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 94–102 (2002)
13. Cooper, G., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning **9**, 309–405 (1992)
14. Cooper, G.F., Yoo, C.: Causal discovery from a mixture of experimental and observational data. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 116–125 (1999)
15. Cox, D.R., Wermuth, N.: Multivariate Dependencies: Models, analysis and interpretation. Chapman and Hall/CRC (1996)
16. Friedman, N., Nachman, I., Peér, D.: Learning Bayesian network structure from massive datasets: the «sparse candidate» algorithm. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 206–215 (1999)
17. Giudici, P., Castelo, R.: Association models for web mining. Data Mining and Knowledge Discovery **5**, 183–196 (2001)
18. Giudici, P., Castelo, R.: Improving Markov chain Monte Carlo model search for data mining. Machine Learning **50**, 127–158 (2003)
19. Harary, F.: Graph Theory. Addison-Wesley, London (1969)
20. Hauser, A., Bühlmann, P.: Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. Journal of Machine Learning Research **13**(1), 2409–2464 (2012)
21. Hauser, A., Bühlmann, P.: Jointly interventional and observational data: estimation of interventional Markov equivalence classes of directed acyclic graphs. Journal of the Royal Statistical Society Series B: Statistical Methodology **77**(1), 291–318 (2015)
22. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning **20**, 197–243 (1995)
23. Kalisch, M., Bühlman, P.: Estimating high-dimensional directed acyclic graphs with the PC-algorithm. Journal of Machine Learning Research **8**(3) (2007)
24. Kalisch, M., Mächler, M., Colombo, D., Maathuis, M.H., Bühlmann, P.: Causal inference using graphical models with the R package pcalg. Journal of Statistical Software **47**, 1–26 (2012)
25. Kočka, T., Castelo, R.: Improved learning of Bayesian networks. In: Breese, J., Koller, D. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 269–276. Morgan Kaufmann (2001)

26. Kočka, T., Bouckaert, R., Studený, M.: On characterizing inclusion of Bayesian networks. In: Breese, J., Koller, D. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 261–268. Morgan Kaufmann (2001)
27. Kuipers, J., Moffa, G., Heckerman, D.: Addendum on the scoring of Gaussian directed acyclic graphical models. Annals of Statistics **42**(4), 1689–1691 (2014)
28. Lauritzen, S.L.: Graphical Models. Oxford University Press, Oxford (1996)
29. Madigan, D., York, J.: Bayesian graphical models for discrete data. International Statistical Review pp. 215–232 (1995)
30. Meek, C.: Graphical models, selecting causal and statistical models. Ph.D. thesis, Carnegie Mellon University (1997)
31. Mukherjee, S., Speed, T.P.: Network inference using informative priors. Proceedings of the National Academy of Sciences **105**(38), 14313–14318 (2008)
32. Nielsen, J.D., Kočka, T., Peña, J.M.: On local optima in learning Bayesian networks. In: Kjærulff, U., Meek, C. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 435–442. Morgan Kaufmann (2003)
33. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, California (1988)
34. Pearl, J.: Causal diagrams for empirical research. Biometrika **82**(4), 669–688 (1995)
35. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge university press (2009)
36. Robinson, R.W.: Counting labeled acyclic digraphs. In: Harary, F. (ed.) New Directions in the Theory of Graphs. pp. 239–273. Academic Press, New York (1973)
37. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics pp. 461–464 (1978)
38. Silander, T., Myllymäki, P.: A simple approach for finding the globally optimal Bayesian network structure. In: Dechter, R., Richardson, T. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 445–452. Morgan Kaufmann (2006)
39. Spirtes, P., Glymour, C.: An algorithm for fast recovery of sparse causal graphs. Social Science Computer Review **9**(1), 62–72 (1991)
40. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search. Springer-Verlag, New York (1993)
41. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning **65**, 31–78 (2006)
42. Verma, T., Pearl, J.: Influence diagrams and d-separation. Tech. Rep. CSD 880052, R-101, Cognitive Systems Laboratory, UCLA (March 1988)
43. Verma, T., Pearl, J.: Equivalence and synthesis of causal models. In: Bonissone, P., Henrion, M., Kanal, L., Lemmer, J. (eds.) Proc. of the Conf. on Uncertainty in Artificial Intelligence. pp. 255–268. Morgan Kaufmann (1990)
44. Whittaker, J.: Graphical Models in Applied Multivariate Statistics. Wiley, New York (1990)
45. Wright, S.: Correlation and causation. Journal of Agricultural Research **20**(7), 557–585 (1921)