Mind Change Optimal Learning of Bayes Net Structure

Oliver Schulte[1], Wei Luo[1], and Russell Greiner[2]

[1] Simon Fraser University, Vancouver-Burnaby, BC V5A 1S6, Canada
{oschulte,wluoa}@cs.sfu.ca
[2] University of Alberta, Edmonton, Alberta T6G 2E8, Canada
greiner@cs.ualberta.ca

Abstract. This paper analyzes the problem of learning the structure of a Bayes net (BN) in the theoretical framework of Gold's learning paradigm. Bayes nets are one of the most prominent formalisms for knowledge representation and probabilistic and causal reasoning. We follow constraint-based approaches to learning Bayes net structure, where learning is based on observed conditional dependencies between variables of interest (e.g., "X is dependent on Y given any assignment to variable Z"). Applying learning criteria in this model leads to the following results. (1) The mind change complexity of identifying a Bayes net graph over variables $\mathbf{V}$ from dependency data is $\binom{|\mathbf{V}|}{2}$, the maximum number of edges. (2) There is a unique fastest mind-change optimal Bayes net learner; convergence speed is evaluated using Gold's dominance notion of "uniformly faster convergence". This learner conjectures a graph if it is the unique Bayes net pattern that satisfies the observed dependencies with a minimum number of edges, and outputs "no guess" otherwise. Therefore we are using standard learning criteria to define a natural and novel Bayes net learning algorithm. We investigate the complexity of computing the output of the fastest mind-change optimal learner, and show that this problem is NP-hard (assuming $\mathrm{P} = \mathrm{RP}$). To our knowledge this is the first NP-hardness result concerning the existence of a uniquely optimal Bayes net structure.

1 Introduction

One of the goals of computational learning theory is to analyze the complexity of practically important learning problems, and to design optimal learning algorithms for them that meet performance guarantees. In this paper, we model learning the structure of a Bayes net as a language learning problem in the Gold paradigm. We apply identification criteria such as mind change bounds [9, Ch. 12.2][20], mind-change optimality [11,12], and text-efficiency (minimizing time or number of data points before convergence) [16,8]. Bayes nets, one of the most prominent knowledge representation formalisms [18,19], are widely used to define probabilistic models in a graphical manner, with a directed acyclic graph (DAG) whose edges link the variables of interest.

We base our model of BN structure learning on an approach known as "constraint-based" learning [5]. Constraint-based learning views a BN structure as a specification of conditional dependencies of the form $X \not\perp\!\!\!\perp Y|\mathbf{S}$, where X and Y are variables of interest and $\mathbf{S}$ is a set of variables disjoint from $\{X, Y\}$. (Read $X \not\perp\!\!\!\perp Y|\mathbf{S}$ as "variable X is dependent on variable Y given values for the variables in the set $\mathbf{S}$".) For example,

N. Bshouty and C. Gentile (Eds.): COLT 2007, LNAI 4539, pp. 187–202, 2007.

a conditional dependence statement represented by a Bayes net may be "father's eye colour is dependent on mother's eye colour given child's eye colour". In this view, a BN structure is a syntactic representation of a dependency relation [18, Sec.3.3]. It is possible for distinct BN structures to represent the same dependency relation; in that case the equivalent BN structures share a partially directed graph known as a *pattern* (defined below), so a BN pattern is a unique syntactic representation of a dependency relation. A dependency relation meets the mathematical definition of a language in the sense of Gold's paradigm, where the basic "strings" are dependence statements of the form "$X \not\perp\!\!\!\perp Y|\mathbf{S}$". We show that in this learning model, the mind change complexity of learning a Bayes net graph for a given set of variables $\mathbf{V}$ is $\binom{|\mathbf{V}|}{2}$—the maximum number of edges in a graph with node set $\mathbf{V}$. Our analysis leads to a characterization of BN learning algorithms that are mind-change optimal. A learner is mind-change optimal if it minimizes the number of mind changes not only globally in the entire learning problem, but also locally in subproblems encountered after receiving some evidence [11,12]; see Section 5. Mind-change optimal BN learners are exactly those that conjecture a BN pattern G only if the pattern is the unique one that satisfies the observed dependencies *with a minimum number of edges*.

Applying Gold's notion of dominance in convergence time [8, p.462], we show that there is a fastest mind-change optimal learner whose convergence time dominates that of all other mind-change optimal learners. The fastest learner is defined as follows: If there is more than one BN pattern G that satisfies the observed dependencies with a minimum number of edges, output "?" (for "no guess"). If there is a unique pattern G that satisfies the observed dependencies with a minimum number of edges, output G. Thus standard identification criteria in Gold's paradigm lead to a natural and novel algorithm for learning BN structure. The technically most complex result of the paper examines the computational complexity of the fastest mind-change optimal BN learner: we show that computing its conjectures is NP-hard (assuming that $\mathrm{P} = \mathrm{RP}$).

Related Work. Many BN learning systems follow the "search and score" paradigm, and seek a structure that optimizes some numeric score [5]. Our work is in the alternative constraint-based paradigm. Constraint-based (CB) algorithms for learning Bayes net structure are a well-developed area of machine learning. Introductory overviews are provided in [5], [15, Ch.10]. The Tetrad system [6] includes a number of CB methods for different classes of Bayes nets. A fundamental difference between existing CB approaches and our model is that the existing methods assume access to an oracle that returns an answer for every query of the form "does $X \not\perp\!\!\!\perp Y|\mathbf{S}$ hold?" In contrast, our model corresponds to the situation of a learner whose evidence (in the form of dependency assertions) grows incrementally over time. Another difference is that existing CB methods assume that their oracle indicates *both* whether two variables are conditionally dependent and whether they are conditionally independent. In language learning terms, the CB method has access to both positive data (dependencies) and negative data (independencies). In our analysis, the learner receives only positive data (dependencies). To our knowledge, our work is the first application of Gold's language learning paradigm to Bayes net learning.

A Bayes net that satisfies a set of given dependencies $\mathcal{D}$ is said to be an I-map for $\mathcal{D}$. We show the NP-hardness of the following problem: for a given set of dependencies $\mathcal{D}$

represented by an oracle O (Section 6), decide whether there is a unique edge minimal I-map G for $\mathcal{D}$, and if so, output G. Bouckaert proved that the problem is NP-hard without the uniqueness condition [2, Lm. 4.5]. However, Bouckaert's proof cannot be adapted for our uniqueness problem, which requires a much more complex reduction. To our knowledge, this is the first NP-hardness result for deciding the existence of a uniquely optimal Bayes net structure for any optimality criterion.

We introduce concepts and results from both learning theory and Bayes net theory in the next section. Section 3 presents and discusses our model of BN structure learning as a language learning problem. Section 4 analyzes the mind change complexity of BN structure learning. Section 5 characterizes the mind-change optimal learning algorithms for this problems and describes the fastest mind-change optimal learner. The final two sections define the problem of computing the output of the fastest mind-change optimal learner and show that the problem is NP-hard.

2 Preliminaries: Language Identification and Bayes Nets

We first introduce general concepts from learning theory, followed by basic definitions from Bayes net theory.

2.1 Language Identification with Bounded Mind Changes

We employ notation and terminology from [10], [13, Ch.1], [16], and [8]. We write $\mathbb{N}$ for the set of natural numbers $\{0, 1, 2, ...\}$. The symbols $\subseteq, \supseteq, \subset, \supset$, and $\emptyset$ respectively stand for subset, superset, proper subset, proper superset, and the empty set. We assume that there is an at most countable set $\mathbb{E}$ of potential evidence items (strings in language learning). A **language** is a subset of $\mathbb{E}$; we write L for a generic language [8, p.449]. A **language learning problem** is a collection of languages; we write $\mathcal{L}$ for a generic collection of languages. A **text** T is a mapping of $\mathbb{N}$ into $\mathbb{E} \cup \{\#\}$, where $\#$ is a symbol not in $\mathbb{E}$. (The symbol $\#$ models pauses in data presentation.) We write $\mathrm{content}(T)$ for the intersection of $\mathbb{E}$ and the range of T. A text T is **for** a language L iff $L = \mathrm{content}(T)$. The initial sequence of text T of length n is denoted by $T[n]$. The set of all finite initial sequences over $\mathbb{E} \cup \{\#\}$ is denoted by SEQ. We also use $\mathrm{SEQ}(\mathcal{L})$ to denote finite initial sequences consistent with languages in $\mathcal{L}$. Greek letters σ and τ range over SEQ. We write $\mathrm{content}(\sigma)$ for the intersection of $\mathbb{E}$ and the range of σ. We write $\sigma \subset T$ to denote that text T extends initial sequence σ; similarly for $\sigma \subset \tau$. A **learner Ψ for** a collection of languages $\mathcal{L}$ is a mapping of $\mathrm{SEQ}(\mathcal{L})$ into $\mathcal{L} \cup \{?\}$. Our term "learner" corresponds to the term "scientist" in [13, Ch.2.1.2]. We say that a learner Ψ **identifies** a language L on a text T for L, if $\Psi(T[n]) = L$ for all but a finitely many n. Next we define identification of a language collection relative to some evidence.

Definition 1. *A learner Ψ for $\mathcal{L}$ **identifies** $\mathcal{L}$ given $\sigma \iff$ for every language $L \in \mathcal{L}$, and for every text $T \supset \sigma$ for L, the learner Ψ identifies L on T.*

Thus a learner Ψ identifies a language collection $\mathcal{L}$ if Ψ identifies $\mathcal{L}$ given the empty sequence Λ. A learner Ψ **changes its mind** at some nonempty finite sequence $\sigma \in \mathrm{SEQ}$ if $\Psi(\sigma) \neq \Psi(\sigma^-)$ and $\Psi(\sigma^-) \neq ?$, where σ^- is the initial segment of σ with σ's last element removed [9, Ch.12.2]. (No mind changes occur at the empty sequence Λ.).

Definition 2. *Let* $\mathrm{MC}(\Psi, T, \sigma)$ *denote the total number of mind changes of* Ψ *on text* T *after sequence* σ *(i.e.,* $\mathrm{MC}(\Psi, T, \sigma) = |\{\tau : \sigma \subset \tau \subset T : \Psi$ *changes its mind at* $\tau\}|$*).*

1. Ψ *identifies* $\mathcal{L}$ ***with mind-change bound*** k *given* $\sigma \iff \Psi$ *identifies* $\mathcal{L}$ *given* σ *and* Ψ *changes its mind at most* k *times on any text* $T \supset \sigma$ *for a language in* $\mathcal{L}$ *after* σ *(i.e., if* $T \supset \sigma$ *extends data sequence* σ *and* T *is a text for any language* $L \in \mathcal{L}$*, then* $\mathrm{MC}(\Psi, T, \sigma) \leq k$*).*
2. *A language collection* $\mathcal{L}$ *is* ***identifiable with mind change bound*** k *given* $\sigma \iff$ *there is a learner* Ψ *such that* Ψ *identifies* $\mathcal{L}$ *with mind change bound k given* σ*.*

2.2 Bayes Nets: Basic Concepts and Definitions

We employ notation and terminology from [19], [18] and [22]. A **Bayes net structure** is a directed acyclic graph $G = (\mathbf{V}, E)$. Two nodes X, Y are **adjacent** in a BN if G contains an edge $X \rightarrow Y$ or $Y \rightarrow X$. The **pattern** $\pi(G)$ of DAG G is the partially directed graph over $\mathbf{V}$ that has the same adjacencies as G, and contains an arrowhead $X \rightarrow Y$ if and only if G contains a triple $X \rightarrow Y \leftarrow Z$ where X and Z are not adjacent. A node W is a **collider on undirected path** p in DAG G if and only if the left and right neighbours of W on p point into W. Every BN structure defines a separability relation between a pair of nodes X, Y relative to a set of nodes $\mathbf{S}$, called **d-separation**. If X, Y are two variables and $\mathbf{S}$ is a set of variables disjoint from $\{X, Y\}$, then $\mathbf{S}$ d-separates X and Y if along every (undirected) path between X and Y there is a node W satisfying one of the following conditions: (1) W is a collider on the path and neither W nor any of its descendants is in $\mathbf{S}$, or (2) W is not a collider on the path and W is in $\mathbf{S}$. We write $(X \perp\!\!\!\perp Y|\mathbf{S})_G$ if X and Y are d-separated by $\mathbf{S}$ in graph G. If two nodes X and Y are not d-separated by $\mathbf{S}$ in graph G, then X and Y are **d-connected** by $\mathbf{S}$ in G, written $(X \not\perp\!\!\!\perp Y|\mathbf{S})_G$. The d-connection relation, or **dependency relation**, for a graph is denoted by $\mathcal{D}_G$, that is, $\langle X, Y, \mathbf{S}\rangle \in \mathcal{D}_G$ iff $(X \not\perp\!\!\!\perp Y|\mathbf{S})_G$. Verma and Pearl proved that two Bayes nets G_1 and G_2 represent the same dependency relation iff they have the same pattern (i.e., $\mathcal{D}_{G_1} = \mathcal{D}_{G_2}$ iff $\pi(G_1) = \pi(G_2)$ [24, Thm. 1]). Thus we use a pattern as a syntactic representation for a Bayes net dependency relation and write G to denote both graphs and patterns unless there is ambiguity. The **statement space** over a set of variables $\mathbf{V}$, denoted by $\mathcal{U}_{\mathbf{V}}$, contains all conditional dependency statements of the form $(X \not\perp\!\!\!\perp Y|\mathbf{S})$, where X, Y are distinct variables in $\mathbf{V}$ and $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$.

Fig. 1 shows a Bayes net from [19, p.15]. In this network, node `wet` is an unshielded collider on the path `sprinkler` − `wet` − `rain`; node `wet` is not a collider on the path `sprinkler` − `wet` − `slippery`. The pattern of the network has the same skeleton, but contains only two edges that induce the collider `wet`. From d-separation we have $(\texttt{sprinkler} \perp\!\!\!\perp \texttt{rain}|\{\texttt{season}\})_G$ and $(\texttt{sprinkler} \not\perp\!\!\!\perp \texttt{rain}|\{\texttt{season}, \texttt{wet}\})_G$. Next we introduce our model of BN structure learning, which associates a language collection $\mathcal{L}_{\mathbf{V}}$ with a given set of variables $\mathbf{V}$; the language collection $\mathcal{L}_{\mathbf{V}}$ comprises all dependency relations defined by Bayes net structures.

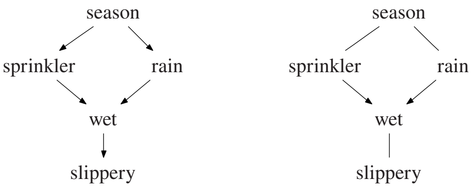

Fig. 1. Sprinkler network and its pattern

Table 1. The correspondence between constraint-based learning of Bayes Nets from conditional dependency data and Gold's language learning model

General Language Learning	Bayes Net Structure Learning
string	conditional dependency statement $X \not\perp\!\!\!\perp Y \vert \mathbf{S}$
language	conditional dependency relation
index	pattern
text	complete dependency sequence

3 Bayes Net Learning with Bounded Mind Changes

This section defines our model of BN structure learning. We discuss the assumptions in the model and compare them to assumptions made in other constraint-based BN learning approaches.

3.1 Definition of the Learning Model

Fix a set of variables $\mathbf{V}$. The evidence item set $\mathbb{E}$ is the statement space $\mathcal{U}_{\mathbf{V}}$. Let $\mathcal{L}_{\mathbf{V}}$ be the **set of BN-dependency relations over variables V** (i.e., $\mathcal{L}_{\mathbf{V}} = \{\mathcal{D}_G : G \text{ is a pattern over } \mathbf{V}\}$). A **complete dependency sequence** T is a mapping of $\mathbb{N}$ into $\mathcal{U}_{\mathbf{V}} \cup \{\#\}$. A dependency sequence T is **for** a dependency relation $\mathcal{D}$ iff $\mathcal{D} = \mathrm{content}(T)$. A Bayes net learning algorithm Ψ maps a finite data sequence σ over $\mathcal{U}_{\mathbf{V}} \cup \{\#\}$ to a pattern G. As Table 1 illustrates, this defines a language learning model, with some changes in terminology that reflect the Bayes net context.

Example. Let G be the DAG in Figure 1. The dependency relation for the graph $\mathcal{D}_G$ contains $\{\ \langle \texttt{season}, \texttt{sprinkler}, \emptyset \rangle,\ \langle \texttt{season}, \texttt{sprinkler}, \{\texttt{rain}\} \rangle,\ \ldots,\ \langle \texttt{sprinkler}, \texttt{rain}, \{\texttt{season}, \texttt{wet}\} \rangle, \langle \texttt{sprinkler}, \texttt{rain}, \{\texttt{season}, \texttt{slippery}\} \rangle \}$. Any text enumerating $\mathcal{D}_G$ is a dependency sequence for $\mathcal{D}_G$.

3.2 Discussion

A Bayes net defines a dependency relation via the d-separation criterion. The motivation for this criterion stems from how a Bayes net represents a probability distribution P. Let P be a joint distribution over variables $\mathbf{V}$. If $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ are three disjoint sets of variables, then $\mathbf{X}$ and $\mathbf{Y}$ are **stochastically independent given S**, denoted by

$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_P$, if $P(\mathbf{X}, \mathbf{Y}|\mathbf{S}) = P(\mathbf{X}|\mathbf{S})\, P(\mathbf{Y}|\mathbf{S})$ whenever $P(\mathbf{S}) > 0$. If $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{S}$ are disjoint sets of nodes in G and $\mathbf{X}$ and $\mathbf{Y}$ are not empty, then $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{S}$ if and only if every pair $\langle X, Y\rangle$ in $\mathbf{X} \times \mathbf{Y}$ is d-separated by $\mathbf{S}$. In constraint-based BN learning, it is common to assume that the probability distribution generating the data of interest has a faithful BN representation [22, Thm.3.2], [19, Ch.2.4].

Definition 3. *Let* $\mathbf{V}$ *be a set of variables,* G *a Bayes net over* $\mathbf{V}$*, and* P *a joint distribution over* $\mathbf{V}$*. Then* G *is* ***faithful to*** P *if* $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_P$ *in* $P \iff (\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_G$ *in* G*.*

Assuming faithfulness, the dependencies in the data can be exactly represented in a Bayes net or a pattern, which is the assumption in our language learning model. It is easy to see that a graph G is faithful to a distribution P if and only if G is faithful with respect to variable pairs, that is, if $(X \not\perp\!\!\!\perp Y|\mathbf{S})_P$ in $P \iff (X \not\perp\!\!\!\perp Y|\mathbf{S})_G$ in G for all variables X, Y. Therefore CB methods focus on conditional dependencies of the form $X \not\perp\!\!\!\perp Y|\mathbf{S}$, which is the approach we follow throughout the paper.

As Gold's paradigm does not specify how linguistic data are generated for the learner, our model does not specify how the observed dependencies are generated. In practice, a BN learner obtains a random sample $\mathbf{d}$ drawn from the operating joint distribution over the variables $\mathbf{V}$, and applies a suitable statistical criterion to decide if a dependency $X \not\perp\!\!\!\perp Y|\mathbf{S}$ holds. One way in which data for our model can be generated from random samples is the following: For every triple $X \not\perp\!\!\!\perp Y|\mathbf{S}$ with $\{X, Y\} \cap \mathbf{S} = \emptyset$, a statistical test is performed with $X \perp\!\!\!\perp Y|\mathbf{S}$ as the null hypothesis. (For small numbers of variables, this is a common procedure in statistics called "all subsets variable selection" [25, p.59].) If the test rejects the null hypothesis, the dependency $X \not\perp\!\!\!\perp Y|\mathbf{S}$ is added to the dependency data; otherwise no conclusion is drawn. Many CB systems also use a statistical test to answer queries to a dependency oracle: given a query "Does $X \not\perp\!\!\!\perp Y|\mathbf{S}$ hold?", the system answers "yes" if the test rejects the hypothesis $X \perp\!\!\!\perp Y|\mathbf{S}$, and "no" otherwise. The assumption that this procedure yields correct results is called the assumption of valid statistical testing [5, Sect.6.2]. Our model is more realistic in two respects. First, the model assumes only that *dependency information* is available, but does not rely on independence data. In fact, many statisticians hold that no independence conclusion should be drawn when a statistical significance test fails to reject an independence hypothesis [7]. Second, our model does not assume that the dependency information is supplied by an oracle all at once, but explicitly considers learning in a setting where more information becomes available as the sample size increases.

Since the set of dependency relations $\mathcal{L}_{\mathbf{V}}$ constitutes a language collection in the sense of the Gold paradigm, we can employ standard identification criteria to analyze this learning problem. We begin by applying a fundamental result in Bayes net theory to determine the mind change complexity of the problem.

4 The Mind Change Complexity of Learning Bayes Net Structure

Following Angluin [1, Condition 3] and Shinohara [21], we say that a class of languages $\mathcal{L}$ has **finite thickness** if the set $\{L \in \mathcal{L} : s \in L\}$ is finite for every string or evidence item $s \in \bigcup \mathcal{L}$. For language collections with finite thickness, their mind change complexity is determined by a structural feature called the inclusion depth [12, Def.6.1].

Definition 4. *Let $\mathcal{L}$ be a language collection and L be a language in $\mathcal{L}$. The* ***inclusion depth*** *of L in $\mathcal{L}$ is the size n of the largest index set $\{L_i\}_{1 \leq i \leq n}$ of distinct languages in $\mathcal{L}$, such that $L \subset L_1 \subset \cdots \subset L_i \subset \cdots \subset L_n$. The* ***inclusion depth*** *of $\mathcal{L}$ is the maximum of the inclusion depths of languages in $\mathcal{L}$.*

The next proposition establishes the connection between inclusion depth and mind change complexity. It follows immediately from the general result for ordinal mind change bounds established in [12, Prop. 6.1].

Proposition 1. *Let $\mathcal{L}$ be a language collection with finite thickness. Then there is a learner Ψ that identifies $\mathcal{L}$ with mind change bound $k \iff$ the inclusion depth of $\mathcal{L}$ is at most k.*

Since we are considering Bayes nets with finitely many variables, the statement space $\mathcal{U}_{\mathbf{V}}$ is finite, so the language collection $\mathcal{L}_{\mathbf{V}}$ containing all BN-dependency relations is finite and therefore $\mathcal{L}_{\mathbf{V}}$ has finite thickness. Hence we have the following corollary.

Corollary 1. *Let $\mathbf{V}$ be a set of variables. There exists a learner Ψ that identifies $\mathcal{L}_{\mathbf{V}}$ with mind change bound $k \iff$ the inclusion depth of $\mathcal{L}_{\mathbf{V}}$ is at most k.*

A fundamental result in Bayes net theory allows us to determine the inclusion depth of a dependency relation in $\mathcal{L}_{\mathbf{V}}$. An edge $A \rightarrow B$ is **covered** in a DAG G if the parents of B are exactly the parents of A plus A itself (see Figure 2). The operation that reverses the direction of the arrow between A and B is a **covered edge reversal**. The following theorem was conjectured by Meek [14] and proven by Chickering [3, Thm.4].

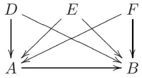

Fig. 2. Edge $A \rightarrow B$ is covered, whereas $D \rightarrow A$ is not covered

Theorem 1 (Meek-Chickering). *Let G and H be two DAGs over the same set of variables $\mathbf{V}$. Then $\mathcal{D}_G \subseteq \mathcal{D}_H \iff$ the DAG H can be transformed into the DAG G by repeating the following two operations: (1) covered edge reversal, and (2) single edge deletion.*

The next corollary characterizes the inclusion depth of the BN dependence relation $\mathcal{D}_G$ for a graph G in terms of a simple syntactic feature of G, namely the number of missing adjacencies.

Corollary 2. *Let $G = (\mathbf{V}, E)$ be a Bayes net structure. Then the inclusion depth of the BN-dependence relation $\mathcal{D}_G$ equals $\binom{|\mathbf{V}|}{2} - |E|$, the number of adjacencies* not *in G. In particular, the totally disconnected network has inclusion depth $\binom{|\mathbf{V}|}{2}$; a complete network has inclusion depth 0.*

Proof. We use downward induction on the number of edges n in graph G. Let $N = \binom{|\mathbf{V}|}{2}$. Base case: $n = N$. Then G is a complete graph, so $\mathcal{D}_G$ contains all dependency statements in the statement space $\mathcal{U}_{\mathbf{V}}$, and therefore has 0 inclusion depth. Inductive step: Assume the hypothesis for $n + 1$ and consider a graph G with n edges. Add an edge to G to obtain a BN G' with $n+1$ edges that is a supergraph of G'. The definition of d-separation implies that $\mathcal{D}_G \subset \mathcal{D}_{G'}$. By inductive hypothesis, there is an inclusion chain $\mathcal{D}_{G'} \subset \mathcal{D}_{G_1} \cdots \subset \mathcal{D}_{G_{N-(n+1)}}$ consisting of BN dependency relations. Hence the inclusion depth of G is at least $N - (n + 1) + 1 = N - n$.

To show that the inclusion depth of G is exactly $N-n$, suppose for contradiction that it is greater than $N - n$. Then there is an inclusion chain $\mathcal{D}_G \subset \mathcal{D}_{H_1} \subset \mathcal{D}_{H_2} \subset \cdots \subset \mathcal{U}_{\mathbf{V}}$ of length greater than $N - n$. So the inclusion depth of $\mathcal{D}_{H_2}$ is at least $N - (n+1)$ and the inclusion depth of $\mathcal{D}_{H_1}$ is at least $N - n$. Hence by inductive hypothesis, the number of edges in H_2 is at most $n+1$ and in H_1 at most n. So at least two of the graphs G, H_1, H_2 have the same number of edges. Without loss of generality, assume that H_1 and H_2 have the same number of edges. Since $\mathcal{D}_{H_1} \subset \mathcal{D}_{H_2}$, Theorem 1 implies that H_1 can be obtained from H_2 with covered edge reversals. But covered edge reversals are symmetric, so we also have $\mathcal{D}_{H_2} \subseteq \mathcal{D}_{H_1}$, which contradicts the choice of H_1 and H_2. So the inclusion depth of $\mathcal{D}_G$ is $N - n$, which completes the inductive proof.

Together with Proposition 1, the corollary implies that the mind change complexity of identifying a Bayes Net structure over variables **V** is given by the maximum number of edges over **V**.

Theorem 2. *For any set of variables* $\mathbf{V}$*, the inclusion depth of* $\mathcal{L}_{\mathbf{V}}$ *is* $\binom{|\mathbf{V}|}{2}$*. So the mind change complexity of identifying the correct Bayes Net structure from dependency data is* $\binom{|\mathbf{V}|}{2}$.

The next section characterizes the BN learning algorithms that achieve optimal mind change performance.

5 Mind-Change Optimal Learners for Bayes Net Structure

We analyze mind-change optimal algorithms for identifying Bayes net structure. The intuition underlying mind-change optimality is that a learner that is efficient with respect to mind changes minimizes mind changes not only globally in the entire learning problem, but also locally in subproblems after receiving some evidence [12,11]. We formalize this idea as in [12, Def.2.3]. If a mind change bound exists for $\mathcal{L}$ given σ, let $\mathrm{MC}_{\mathcal{L}}(\sigma)$ be the least k such that $\mathcal{L}$ is identifiable with k mind changes given σ. For example, given a sequence σ of dependencies, let $G = (\mathbf{V}, E)$ be a BN that satisfies the dependencies in σ with a minimum number of edges. Then the mind change complexity $\mathrm{MC}_{\mathcal{L}_{\mathbf{V}}}(\sigma)$ is $\binom{|\mathbf{V}|}{2} - |E|$. Mind change optimality requires that a learner should succeed with $\mathrm{MC}_{\mathcal{L}}(\sigma)$ mind changes after each data sequence σ.

Definition 5 (based on Def.2.3 of [12]). *A learner* Ψ *is* ***strongly mind-change optimal*** *(SMC-optimal) for* $\mathcal{L}$ *if for all data sequences* σ *the learner* Ψ *identifies* $\mathcal{L}$ *given* σ *with at most* $\mathrm{MC}_{\mathcal{L}}(\sigma)$ *mind changes.*

The next proposition characterizes SMC-optimal learners for language collections with finite inclusion depth. It follows from the general characterization of SMC-optimal learners for all language collections established in [12, Prop.4.1].

Proposition 2. *Let Ψ be a learner that identifies a language collection $\mathcal{L}$ with finite inclusion depth. Then Ψ is SMC-optimal for $\mathcal{L}$ if and only if for all data sequences σ: if $\Psi(\sigma) \neq ?$, then $\Psi(\sigma)$ is the unique language with the largest inclusion depth for σ.*

Applying the proposition to Bayes net learners yields the following corollary.

Corollary 3. *Let Ψ be a Bayes net learner that identifies the correct Bayes net pattern for a set of variables $\mathbf{V}$. The learner Ψ is SMC-optimal for $\mathcal{L}_{\mathbf{V}} \iff$ for all dependency sequences σ, if the output of Ψ is not ?, then Ψ outputs a uniquely edge-minimal pattern for the dependencies $\mathcal{D} = \mathrm{content}(\sigma)$.*

It is easy to implement a slow SMC-optimal BN learner. For example, for a given set of dependencies $\mathcal{D}$ it is straightforward to check if there is a pattern G that covers exactly those dependencies (i.e., $\mathcal{D}_G = \mathcal{D}$). So an SMC-optimal learner could output a pattern G if there is one that matches the observed dependencies exactly, and output ? otherwise. But such a slow learner requires exponentially many dependency statements as input. There are SMC-optimal learners that produce a guess faster; in fact, using Gold's notion of "uniformly faster", we can show that there is a unique fastest SMC-optimal learner. Gold proposed the following way to compare the convergence speed of two learners [8, p. 462].

Definition 6. *Let $\mathcal{L}$ be a language collection.*

1. *The convergence time of a learner Ψ on text T is defined as $\mathrm{CP}(\Psi, T) \equiv$ the least time m such that $\Psi(T[m]) = \Psi(T[m'])$ for all $m' \geq m$.*
2. *A learner Ψ identifies $\mathcal{L}$ uniformly faster than learner $\Phi \iff$*
   (a) for all languages $L \in \mathcal{L}$ and all texts T for L, we have $\mathrm{CP}(\Psi, T) \leq \mathrm{CP}(\Phi, T)$, and
   (b) for some language $L \in \mathcal{L}$ and some text T for L, we have $\mathrm{CP}(\Psi, T) < \mathrm{CP}(\Phi, T)$.

For a language collection $\mathcal{L}$ with finite inclusion depth, Proposition 2 implies that if there is no language L that uniquely maximizes inclusion depth given σ, then a learner that is SMC-optimal outputs ? on σ. Intuitively, the fastest SMC-optimal learner delays making a conjecture no longer than is necessary to meet this condition. Formally, this learner is defined as follows for all sequences $\sigma \in \mathrm{SEQ}(\mathcal{L})$:

$$\Psi^{\mathcal{L}}_{\mathrm{fast}}(\sigma) = \begin{cases} ? & \text{if no language uniquely maximizes inclusion depth given } \sigma \\ L & \text{if } L \in \mathcal{L} \text{ uniquely maximizes inclusion depth given } \sigma. \end{cases}$$

The next observation asserts that $\Psi^{\mathcal{L}}_{\mathrm{fast}}$ is the fastest SMC-optimal method for $\mathcal{L}$.

Observation 1. *Let $\mathcal{L}$ be a language collection with finite inclusion depth. Then $\Psi^{\mathcal{L}}_{\mathrm{fast}}$ is SMC-optimal and identifies $\mathcal{L}$ uniformly faster than any other SMC-optimal learner for $\mathcal{L}$.*

Proof. The proof is a variant of standard results on text-efficiency (e.g., [13, Ch.2.3.3]) and is omitted for space reasons.

Observation 1 leads to the following algorithm for identifying a BN pattern.

Corollary 4. *Let* $\mathbf{V}$ *be a set of variables. For a given sequence of dependencies* σ*, the learner* $\Psi^{\mathbf{V}}_{\mathrm{fast}}$ *outputs ? if there is more than one edge-minimal pattern that covers the dependencies in* σ*, and otherwise outputs a uniquely edge-minimal pattern for the dependencies* $\mathcal{D} = \mathrm{content}(\sigma)$. *The learner* $\Psi^{\mathbf{V}}_{\mathrm{fast}}$ *is SMC-optimal and identifies the correct pattern uniformly faster than any other SMC-optimal BN structure learner.*

The remainder of the paper analyzes the run-time complexity of the $\Psi^{\mathbf{V}}_{\mathrm{fast}}$ method; we show that computing the output of the learner is NP-hard (assuming that $\mathrm{P} = \mathrm{RP}$).

6 Computational Complexity of Fast Mind-Change Optimal Identification of Bayes Net structure

This section considers the computational complexity of implementing the fastest SMC-optimal learner $\Psi^{\mathbf{V}}_{\mathrm{fast}}$. We describe the standard approach of analyzing the complexity of constraint-based learners in the Bayes net literature and state some known results from complexity theory for background.

As with any run-time analysis, an important issue is the representation of the input to the algorithm. The most straightforward approach for our learning model would be to take the input as a list of dependencies, and the input size to be the size of that list. However, in practice CB learners do not receive an explicitly enumerated list of dependencies, but rather they have access to a dependency oracle (cf. Section 3.2). Enumerating relevant dependencies through repeated queries is part of the computational task of a CB learner. Accordingly, the standard complexity analysis takes a dependency oracle and a set of variables as the input to the learning algorithm (e.g., [4, Def.12],[2]).

Definition 7. *A dependency oracle* O *for a variable set* $\mathbf{V}$ *is a function that takes as input dependency queries from the statement space* $\mathcal{U}_{\mathbf{V}}$ *and returns, in constant time, either "yes" or "?".*

The dependency relation associated with oracle O is given by $\mathcal{D}_O = \{X \not\perp\!\!\!\perp Y|\mathbf{S} \in \mathcal{U}_{\mathbf{V}} : O$ returns "yes" on input $X \not\perp\!\!\!\perp Y|\mathbf{S}\}$. We note that our model of learning Bayes net structure can be reformulated in terms of a sequence of oracles: Instead of a complete sequence of dependency statements for a dependence relation $\mathcal{D}_G$, the learner could be presented with a sequence of dependency oracles $O_1, O_2, \ldots, O_n, \ldots$ such that $\mathcal{D}_{O_i} \subseteq \mathcal{D}_{O_{i+1}}$ and $\bigcup_{i=1}^{\infty} \mathcal{D}_{O_i} = \mathcal{D}_G$. The mind change and convergence time results remain the same in this model.

We will reduce the problem of computing the output of the fastest mind change optimal learner $\Psi^{\mathbf{V}}_{\mathrm{fast}}$ to deciding the existence of a unique exact cover by 3-sets.

UEC3Set
Instance A finite set X with $|X| = 3q$ and a collection C of 3-element subsets of X.
Question Does C contain a unique *exact cover* for X, that is, a unique subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?

We apply the following well-known result. The class RP comprises the decision problems that can be decided in polynomial time with a randomized algorithm [17, Def.11.1].

Proposition 3. *A polynomial time algorithm for* UEC3SET *yields a polynomial time algorithm for the satisfiability problem* SAT *provided that* $\mathrm{P} = \mathrm{RP}$. *So* UEC3SET *is NP-hard under that assumption.*

The proposition follows from the famous theorem of Valiant and Vazirani that gives a probabilistic reduction of SAT to UNIQUE SAT [23]. Standard reductions show that UNIQUE SAT reduces to UEC3SET. Computing the conjectures of the learner $\Psi^{\mathbf{V}}_{\mathrm{fast}}$ poses the following computational problem.

UNIQUE MINIMAL I-MAP

Input A set of variables $\mathbf{V}$ and a dependency oracle O for $\mathbf{V}$.

Output If there is a *unique* DAG pattern G that covers the dependencies in O with a minimal number of edges, output G. Otherwise output ?.

This is a function minimization problem; the corresponding decision problem is the following.

UNIQUE I-MAP

Instance A set of variables $\mathbf{V}$, a dependency oracle O for $\mathbf{V}$, and a bound k.

Question Is there a DAG pattern G such that: G covers the dependencies in O, every other DAG pattern G' covering the dependencies in O has more edges than G, and G has at most k edges?

Clearly an efficient algorithm for the function minimization problem yields an efficient algorithm for UNIQUE I-MAP. We will show that UNIQUE I-MAP is NP-hard, assuming that $\mathrm{P} = \mathrm{RP}$. Let $\leq_P$ denote polynomial-time many-one reducibility.

Theorem 3. UEC3SET $\leq_{\mathrm{P}}$ UNIQUE I-MAP $\leq_{\mathrm{P}}$ UNIQUE MINIMAL I-MAP. *So* UNIQUE MINIMAL I-MAP *is NP-hard provided that* $\mathrm{P} = \mathrm{RP}$.

Proof. We give a reduction from UEC3SET to UNIQUE I-MAP. Consider an instance of UEC3SET with sets universe U of size $|U| = 3m$, and $c_1, .., c_p$, where $|c_i| = 3$ for $i = 1, .., p$ and $U = \cup_{i=1}^{m} c_i$. Define the following set V of variables.

1. For every set c_i, a *set variable* C_i.
2. For every element x_j of the universe U, a *member variable* X_j.
3. A *root variable* R.

Set the bound $k = 3p + m$. The following program M implements a dependency oracle O over the variables V, in time polynomial in the size of the given UEC3SET instance.

Definition of Dependency Oracle

Input A dependency query $V_1 \not\perp\!\!\!\perp V_2 | \mathbf{S}$.

Output Oracle Clauses

1. If $V_1 = C_i, V_2 = X_j$, and $x_j \in c_i$, then return "dependent".

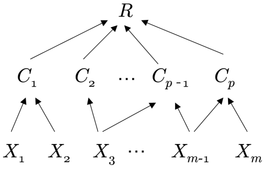

Fig. 3. The basic graph for the NP-hardness proof. A set cover of size m corresponds to m edges of the form $C \rightarrow R$.

2. If $V_1 = X_i, V_2 = X_j$, and there is a set $c_k \supseteq \{x_i, x_j\}$ such that $C_k \in \mathbf{S}$, then return "dependent".
3. If $V_1 = R, V_2 = X_j, \mathbf{S} = \emptyset$ then return "dependent".
4. If $V_1 = R, V_2 = X_j, |\mathbf{S}| = 1$, and $\mathbf{S} \neq \{C\}$ where $x_j \in c$, then return "dependent".
5. In all other cases, return ?.

We argue that there is a unique exact set cover for an instance $\langle U, \{c_i\}\rangle$ iff there is a unique I-map with at most k edges for O. So if there were a polynomial time algorithm A for UNIQUE I-MAP, we could solve the UEC3SET instance in polynomial time by using the program M to "simulate" the oracle O and use A to solve the corresponding instance of UNIQUE I-MAP. Our proof strategy is as follows. The *basic graph* for O is the following DAG B: (1) for every two variables X_j, C_i, the graph contains an arrow $X_j \rightarrow C_i$ iff $x_j \in c_i$, and (2) for every variable C_i, there is an arrow $C_i \rightarrow R$. The basic graph is also a pattern because all arrows correspond to unshielded colliders; see Figure 3. We show that if there is a unique I-map G for O with at most k edges, then G is a subgraph of the basic graph, with possibly edges $C_i \rightarrow R$ missing for some sets c_i, such that the set of variables $\{C_1, C_2, ..., C_m\}$ with the edge $C_i \rightarrow R$ in G corresponds to an exact cover $\{c_1, .., c_m\}$. Conversely, any unique exact cover corresponds to a subgraph of the basic graph in the same manner. For reasons of space, we just illustrate most of the following assertions rather than giving full proofs. It is easiest to consider separately the constraints imposed by each clause of M. Let $\mathcal{D}_i$ be the set of dependencies corresponding to Clause i. For example, $D_1 = \{\langle C_i, X_j, \mathbf{S}\rangle : x_j \in c_i\}$.

Assertion 1. *Let DAG G be an I-map for $\mathcal{D}_1$. Then any two variables X and C are adjacent whenever $x \in c$.*

Assertion 2. *Let DAG G be an I-map for $\mathcal{D}_1 \cup \mathcal{D}_2$, and suppose that x_i, x_j are two elements of a set c. Then X_i and X_j are adjacent in G, or G contains a component $X_i \rightarrow C \leftarrow X_j$.*

Clause 3 requires that every member variable X be d-connected to the root variable. The intuition is that the basic graph B contains the most edge-efficient way to achieve the connection because with just one edge $C \rightarrow R$ the graph d-connects three member variables at once. We show that any I-map for $\mathcal{D}_3$ can be transformed into a subgraph of B without increasing the number of edges. We begin by establishing that in an I-map

G of $\mathcal{D}_3$, all arcs originating in the root variable R can be reversed with the result G' still an I-map of $\mathcal{D}_3$.

Assertion 3. *Let DAG G be an I-map of $\mathcal{D}_3$. Let G' be the graph obtained by reversing all edges of the form $R \rightarrow V$. Then G' is an I-map of $\mathcal{D}_3$.*

Illustration: Suppose G contains a component $R \rightarrow X \rightarrow X'$. Reverse the edge $R \rightarrow X$ to obtain G'. Consider the d-connecting path $R \rightarrow X \rightarrow X'$ in G. We can replace the edge $R \rightarrow X$ by $R \leftarrow X$ in G' without introducing additional colliders, so d-connection still holds. The next assertion shows that inductively, all nodes can be oriented towards R.

Assertion 4. *Let DAG G be an I-map of $\mathcal{D}_3$, with some node A an ancestor of R. Let G' be the graph obtained by reversing all edges of the form $A \rightarrow V$ where V is not an ancestor of R. Then G' is an I-map of $\mathcal{D}_3$.*

Illustration: Suppose G contains a component $X' \leftarrow X \rightarrow C \rightarrow R$. Reverse the edge $X' \leftarrow X$ to obtain G'. Consider the d-connecting path $X' \leftarrow X \rightarrow C \rightarrow R$ in G. In any such directed path in G' we can replace the edge $X' \leftarrow X$ by $X' \rightarrow X$ in G' without introducing additional colliders, so d-connection still holds.

Assertion 5. *Let DAG G be an I-map of $\mathcal{D}_3$. Suppose that for some node V, there are two directed paths $V \rightarrow U_1 \rightarrow \cdots \rightarrow U_p \rightarrow R$ and $V \rightarrow W_1 \rightarrow \cdots \rightarrow W_q \rightarrow R$. Let G' be the graph obtained from G by deleting the edge $V \rightarrow U_1$. Then G' is an I-map of $\mathcal{D}_3$.*

Illustration: Suppose G contains two paths $X \rightarrow C \rightarrow R$ and $X \rightarrow X' \rightarrow R$. Delete the edge $X \rightarrow X'$ to obtain G'. Then X remains d-connected to R. In general, a d-connecting path to R in G using the edge $X \rightarrow X'$ can be "rerouted" via either X or X'.

For a DAG G, let $\text{sets}(G) = \{C : C \text{ is adjacent to } R \text{ in } G\}$ comprise all set variables adjacent to R; these set variables are *covered*. A member variable X is *covered* in G if there is a covered set variable C such that $x \in c$. The *covered component* of G consists of the root variable R, and the covered set and member variables of G (so the covered component is $\{R\} \cup \text{sets}(G) \cup \{X : \exists C \in \text{sets}(G) \text{ s.t. } x \in c\}$). A DAG G is *normally directed* if all covered components of G are ancestors of the root variable R. By Assertion 4 we can normally direct every DAG G and still satisfy the dependencies in $\mathcal{D}_3$.

Assertion 6. *Let DAG G be a normally directed I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$. Suppose that G contains an adjacency $V - V'$ where V is covered in G and V' is not. Unless $V - V' = X \rightarrow C$ for $x \in c$, there is a normally directed I-map G' of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ such that V' is covered in G', all covered variables in G are covered in G', and G' has no more edges than G.*

Illustration: Suppose G contains an edge $X \rightarrow C$ where X is not covered, and a path $X \rightarrow X' \rightarrow C' \rightarrow R$. Add the edge $C \rightarrow R$ and delete the edge $X \rightarrow X'$ to obtain G'. Then X is d-connected to R via C. In general, a d-connecting path in G using the edge $X \rightarrow X'$ can be "rerouted" via either X or X'.

Assertion 7. *Suppose that DAG G is an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ and not all member variables X are covered in G. Then there is an I-map G' of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ that covers all member variables such that G' has no more edges than G, and* $\mathrm{sets}(G') \supset \mathrm{sets}(G)$.

Illustration: Suppose that X is uncovered, and that G contains an edge $X \to C$. Since X is not covered, the edge $C \to R$ is not in G. Since G covers $\mathcal{D}_3$, the variable X must be d-connected to the root variable R; suppose that G contains an edge $X \to R$. We can add an edge $C \to R$ to obtain G^* without losing any d-connection. Now there are two directed paths connecting X to R, so by Assertion 5 deleting the edge $X \to R$ yields a graph G' with the same number of edges as G that is still an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$.

Assertion 8. *No I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ has fewer than k edges.*

Proof: Let G be an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$. By Assertion 1, every I-map G of $\mathcal{D}_1 \cup \mathcal{D}_2$ contains $3p$ edges connecting each member variable with the set variables for the sets containing it. By Assertion 6 we can transform G into a graph G' such that G' is an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, covers all its member variables, and has the same number of edges as G. Thus $\mathrm{sets}(G')$ is a set cover for U, and so the size of $\mathrm{sets}(G')$ is at least m, which means that we have at least m edges connecting the root variable R to set variables. Hence overall G' and hence G has $k = 3p + m$ edges.

Assertion 9. *Let DAG G be an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ with k edges. Then for every uncovered member variable X of G, there is exactly one undirected path from X to R in G.*

Illustration: Suppose that G contains an edge $X \to R$ and a path $X \to X' \to C' \to R$ where X is not covered. Then as in Assertion 5, we can delete the edge $X \to R$ to obtain a graph with fewer than k edges that is still an I-map of $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$. But this contradicts Assertion 8. The final assertion adds the constraints of Clause 4.

Assertion 10. *Let DAG G be an I-map of O with k edges. Then G is normally directed, every member variable in G is covered, and* $\mathrm{sets}(G)$ *is an exact set cover of U.*

An exact set cover corresponds to a unique normally directed I-map for the dependency oracle O with $k = 3p + m$ edges (the I-map contains m edges $C \to R$ for each set c in the cover). Conversely Assertion 10 implies that every I-map for O with k edges corresponds to a unique exact set cover. Hence there is a 1-1 and onto correspondence between exact set covers and I-maps for O.

7 Conclusion

This paper applied learning-theoretic analysis to a practically important learning problem: identifying a correct Bayes net structure. We presented a model of this task in which learning is based on conditional dependencies between variables of interest. This model fits Gold's definition of a language learning problem, so identification criteria from Gold's paradigm apply. We considered mind-change optimality and text efficiency. The mind change complexity of identifying a Bayes net over variable set $\mathbf{V}$ is $\binom{|\mathbf{V}|}{2}$, the maximum number of edges in a graph with node set $\mathbf{V}$. There is a unique mind-change optimal learner $\Psi_{\text{fast}}^{\mathbf{V}}$ whose convergence time dominates that of all other mind-change optimal learners. This learner outputs a BN pattern G if G is the unique graph satisfying the observed dependencies with a minimum number of edges; otherwise $\Psi_{\text{fast}}^{\mathbf{V}}$ outputs ? for "no guess". In many language learning problems, it is plausible to view the mind change complexity of a language as a form of simplicity [12, Sec.4]. Our results establish that the mind-change based notion of simplicity for a Bayes net graph G is the inclusion depth of G, which is measured by the number of edges absent in G. Using the number of edges as a simplicity criterion to guide learning appears to be a new idea in Bayes net learning research.

The technically most complex result of the paper shows that an exact implementation of the unique mind-change optimal learner $\Psi_{\text{fast}}^{\mathbf{V}}$ is NP-hard because determining whether there is a uniquely simplest (edge-minimal) Bayes net for a given set of dependencies is NP-hard. To our knowledge, this is the first NP-hardness result for deciding the existence of a uniquely optimal Bayes net structure by any optimality criterion.

Acknowledgements

This research was supported by NSERC discovery grants to the first and third author. We are indebted to Josh Buresh-Oppenheim for discussions of complexity theory. Kevin T. Kelly suggested considering Bayes net learning based on conditional dependencies rather than independencies. We are indebted to the anonymous COLT reviewers for helpful comments. The Alberta Ingenuity Centre for Machine Learning provided valuable support.

References

1. Angluin, D.: Inductive inference of formal languages from positive data. I&C 45, 117–135 (1980)
2. Bouckaert, R.: Bayesian belief networks: from construction to inference. PhD thesis, U. Utrecht (1995)
3. Chickering, D.: Optimal structure identification with greedy search. JMLR 3, 507–554 (2003)
4. Chickering, D., Heckerman, D., Meek, C.: Large-sample learning of bayesian networks is NP-hard. JMLR 5, 1287–1330 (2004)
5. Cooper, G.: An overview of the representation and discovery of causal relationships using bayesian networks. In: Computation, Causation, and Discovery, pp. 4–62 (1999)
6. Scheines, R., et al.: TETRAD 3 User's Manual. CMU (1996)
7. Giere, R.: The significance test controversy. BJPS 23(2), 170–181 (1972)
8. Gold, E.M.: Language identification in the limit. Info. and Cont. 10(5), 447–474 (1967)
9. Jain, S., Osherson, D., Royer, J., Sharma, A.: Systems That Learn, 2nd edn. MIT Press, Cambridge (1999)
10. Jain, S., Sharma, A.: Mind change complexity of learning logic programs. TCS 284, 143–160 (2002)
11. Luo, W., Schulte, O.: Mind change efficient learning. In: Auer, P., Meir, R. (eds.) COLT 2005. LNCS (LNAI), vol. 3559, pp. 398–412. Springer, Heidelberg (2005)

12. Luo, W., Schulte, O.: Mind change efficient learning. Info. & Comp. 204, 989–1011 (2006)
13. Martin, E., Osherson, D.N.: Elements of Scientific Inquiry. MIT Press, Cambridge (1998)
14. Meek, C.: Graphical Models: Selecting causal and stat. models. PhD thesis, CMU (1997)
15. Neapolitan, R.E.: Learning Bayesian Networks. Pearson Education (2004)
16. Osherson, D., Stob, M., Weinstein, S.: Systems that learn. MIT Press, Cambridge (1986)
17. Papadimitriou, C.H.: Computational complexity. Addison-Wesley, London (1994)
18. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kauffmann, San Francisco (1988)
19. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (2000)
20. Putnam, H.: Trial and error predicates and the solution to a problem of mostowski. JSL 30(1), 49–57 (1965)
21. Shinohara, T.: Inductive inference of monotonic formal systems from positive data. New Gen. Comp. 8(4), 371–384 (1991)
22. Spirtes, P., Glymour, C., Scheines, R.: Causation, prediction, and search. MIT Press, Cambridge (2000)
23. Valiant, L., Vazirani, V.: NP is as easy as detecting unique solutions. TCS 47, 85–93 (1986)
24. Verma, T., Pearl, J.: Equiv. and synth. of causal models. In: UAI'90, pp. 220–227 (1990)
25. Zucchini, W.: An introduction to model selection. J. Math. Psyc. 44, 41–61 (2000)