A Characterization of Single-Link Search in Learning Belief Networks

Y. Xiang

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2, yxiang@cs.uregina.ca

Abstract. One alternative to manual acquisition of belief networks from domain experts is automatic learning of these networks from data. Common algorithms for learning belief networks employ a single-link lookahead search. It is unclear, however, what types of domain models are learnable by such algorithms and what types of models will escape. We conjecture that these learning algorithms that use a single-link search are specializations of a simple algorithm which we call LIM. We put forward arguments that support such a conjecture, and then provide an axiomatic characterization of models learnable by LIM.
The characterization coupled with the conjecture identifies models that are definitely learnable and definitely unlearnable by a class of learning algorithms. It also identifies models that are highly likely to escape these algorithms. Research to formally prove the conjecture is ongoing.

Keywords: knowledge acquisition, learning, knowledge discovery.

1 Introduction

Belief networks [9, 5] provide a coherent and effective framework for building knowledge based systems that must reason with uncertain knowledge. Acquisition of such networks by manual elicitation from experts, however, has been quite time consuming. One alternative to elicitation is to construct such networks by automatic learning from data, which has been an active research area in recent years [3, 7, 1, 12, 2].

Common algorithms for learning belief networks (as referenced above) use a single-link lookahead search to select candidate network structures. It is unclear, however, what types of (probabilistic) domain models are learnable by such algorithms and what types of models will escape. We investigate this issue by finding a generalization of these algorithms and then characterizing the generalized algorithm axiomatically. In this paper, we present our progress in this quest.

In particular, we present a simple algorithm which we call LIM. We provide an axiomatic characterization of models learnable by LIM. We conjecture that LIM is a generalization of a class of common algorithms, and put forward arguments that support this conjecture.

The characterization coupled with the conjecture identifies models that are definitely learnable, definitely unlearnable by the whole class of algorithms, and models that are highly likely to escape these algorithms. It also suggests directions for improving these algorithms and provides a basis to analysis of the new learning algorithms. Our current research effort is directed toward formally proving the conjecture.

In Section 2, we overview the necessary background. We introduce the algorithm LIM in Section 3. The generalization conjecture is presented and argued in Section 4. In Section 5, we show that common concepts in belief network literature are inadequate to characterize domain models learnable by LIM. An axiomatic characterization is presented in Section 6 and its implications are discussed in Section 7.

2 Background

Let N be a set of discrete variables in a problem domain. Each variable is associated with a set of possible values. A *configuration* or a *tuple* of $N' \subseteq N$ is an assignment of values to every variable in N'. A *probabilistic domain model* (PDM) over N determines the probability of every tuple of N' for each $N' \subseteq N$. For three disjoint sets X, Y and Z of variables, X and Y are independent given Z if $P(X|Y,Z) = P(X|Z)$ *whenever* $P(Y,Z) > 0$. We denote the conditional independence relation by $I(X,Z,Y)$. A PDM satisfies the following axioms [9] where $\Longrightarrow$ stands for implication:

Symmetry $I(X,Z,Y) \Longrightarrow I(Y,Z,X)$.
Decomposition $I(X,Z,Y \cup W) \Longrightarrow I(X,Z,Y)\ \&\ I(X,Z,W)$.
Weak Union $I(X,Z,Y \cup W) \Longrightarrow I(X,Z \cup W,Y)$.
Contraction $I(X,Z,Y)\ \&\ I(X,Z \cup Y,W) \Longrightarrow I(X,Z,Y \cup W)$.

If the PDM is strictly positive, then the following also holds:

Intersection $I(X,Z \cup W,Y)\ \&\ I(X,Z \cup Y,W) \Longrightarrow I(X,Z,Y \cup W)$.

For disjoint subsets X, Y and Z of nodes in a graph G, we use $< X|Z|Y >_G$ to denote that nodes in Z *graphically separate* nodes in X and nodes in Y. When G is undirected, $< X|Z|Y >_G$ denotes that nodes in Z intercept all paths between X and Y. When G is a directed acyclic graph (DAG), graphical separation is defined by *d-separation* [9]. A graph G is an *I-map* of a PDM over N if there is an one-to-one correspondence between nodes of G and variables in N such that for all disjoint subsets X, Y and Z of N, $< X|Z|Y >_G \Longrightarrow I(X,Z,Y)$. G is a *D-map* if whenever $I(X,Z,Y)$ holds, X and Y are separated by Z in G, i.e., $< X|Z|Y >_G \Longleftarrow I(X,Z,Y)$. G is a *P-map* if it is both an I-map and a D-map. G is a *minimal* I-map if no link can be removed such that the resultant graph is still an I-map.

A dependency model M (may or may not be a PDM) with an undirected P-map is called a *graph-isomorph* [9]. M is a graph-isomorph iff it satisfies Symmetry, Decomposition, Intersection and the following axioms, where $v \in N$:

Strong Union $I(X,Z,Y) \Longrightarrow I(X,Z \cup W,Y)$.
Transitivity $I(X,Z,Y) \Longrightarrow I(X,Z,v)$ *or* $I(v,Z,Y)$.

A dependency model M with a P-map that is a DAG is called a *DAG isomorph* [9]. A DAG isomorph satisfies Symmetry, Decomposition, Intersection, Weak Union, Contraction and the following axioms, where $x, y, z, v \in N$:

Composition $I(X, Z, Y)$ & $I(X, Z, W) \Longrightarrow I(X, Z, Y \cup W)$.
Weak Transitivity $I(X, Z, Y)$ & $I(X, Z \cup \{v\}, Y) \Longrightarrow I(X, Z, v)$ *or* $I(v, Z, Y)$.
Chordality $I(x, \{v, z\}, y)$ & $I(v, \{x, y\}, z) \Longrightarrow I(x, v, y)$ *or* $I(x, z, y)$.

A belief network consists of a graph structure and a jpd factorized according to the structure. Commonly used structures are DAGs for Bayesian networks (BNs) and chordal graphs for decomposable Markov networks (DMNs) [15]. Common algorithms for learning belief networks [3, 7, 1, 12, 2] start with an empty graph (no links). Links are added to the current graph one at a time (the single-link lookahead search). All graphs differing from the current graph by a single link are evaluated according to a scoring metric before the one with the highest score is adopted (the greedy search).

3 LIM: A Simple Learning Algorithm

We shall take it for granted that the ideal outcome of an algorithm for learning belief networks is an approximate minimal I-map of the data generating PDM (see [9] for arguments for the minimal I-map). In order to characterize models learnable by algorithms using the single-link lookahead search, we propose an algorithm that captures the basic features of these algorithms, which we shall refer to as LIM for `Learning I-Maps`.

LIM is equipped with a test whether $P(X|Y, Z) = P(X|Z)$ holds (equivalent to $I(X, Y, Z)$) for three disjoint subsets of variables X, Y and Z. Clearly, if we allow such test to be performed for arbitrary X, Y and Z, then LIM will be able to learn an I-map of any PDM. Unfortunately, the complexity of LIM will be exponential. We therefore restrict LIM such that the test is only performed based on the currently learned graph in the following manner:

LIM starts with an empty graph G. It systematically selects a link $\{x, y\}$ not contained in G such that one of the following two cases is true:

1. x and y are contained in different components of G.
2. Every node (at least one) adjacent to both x and y is adjacent to every other such node, and these nodes intercept every path between x and y.

We shall call the links that satisfy the above conditions *type 1* and *type 2* links, respectively. In Figure 1, the missing link (b, c) is a type 1 link, and (a, d) and (d, g) are type 2 links. For a type 1 link, LIM tests if $P(x|y) = P(x)$ (equivalent to $I(x, \phi, y)$). For a type 2 link, LIM tests if $P(x|y, C) = P(x|C)$ (equivalent to $I(x, C, y)$), where C is the set of nodes adjacent to both x and y. If the test is negative, then the link $\{x, y\}$ is added to the current graph. LIM repeats the above until no type 1 or type 2 links can be added.

The following Theorem shows that LIM actually returns a chordal graph and therefore learns a DMN. As the learned DMN will be an approximation of the data-generating PDM, we do not require the structure of a DMN to be a minimal

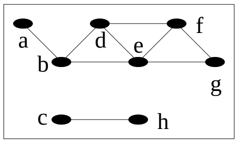

Fig. 1. Illustration of type 1 and type 2 links

I-map of the PDM as required in some literature (e.g., in [9]). DMNs are closely related to BNs but are simpler to study for the purpose of this analysis[1].

Theorem 1 *For any PDM, LIM returns a chordal graph on termination.*

Proof:

We prove by induction on the number i of links in the learned graph. LIM starts with an empty graph ($i = 0$) which is chordal. We assume that when LIM learns $i = k \geq 0$ links, the graph G is chordal.

When LIM learns the $k + 1$'th link $\{x, y\}$, the link must be added to G as either type 1 or type 2. Denote the new graph by G'. If $\{x, y\}$ is type 1, it connects two components of G. Since G is chordal by assumption, each component of G is chordal. When two components are connected by a single-link, the resultant new component is also chordal. Hence G' is chordal.

We now show by contradiction that G' is chordal if $\{x, y\}$ is added as type 2. Assume that G' is not chordal. Then there must be a cycle of length > 3 without a chord. Since G is chordal, the new link $\{x, y\}$ must be in the cycle. There must be at least two other nodes v and w in the cycle such that $x - v - ... - w - y$ form a simple path in G, $\{x, w\}$ is not in G, $\{v, y\}$ is not in G, and every other node on the path is adjacent to neither x nor y. Now we have found a path between x and y in G on which none of the nodes is adjacent to both x and y. This contradicts the assumption that $\{x, y\}$ is a type 2 link. □

4 The Generalization Conjecture

LIM can be viewed as a generalization of several commonly used algorithms for learning belief network structures [3, 7, 1, 12, 2]. We formalize this view as the following conjecture.

Conjecture 1 *Given any PDM M, if common algorithms can learn an I-map of M, then so can LIM.*

LIM appears to differ from common algorithms in many aspects. We argue for Conjecture 1 as follows:

First, we consider the scoring metric. LIM is equipped with a conditional independence (CI) test restricted by the currently learned graph structure. Although a test similar to what is used in LIM was used by Rebane and Pearl

[1] See [8], for example, for how testing d-separation [9] in a DAG D (a BN structure) can be performed in a simpler way in an undirected graph converted from D.

[10], most common algorithms use entropy [3, 12], cross entropy [15], description length (MDL) [7], and Bayesian score [1, 2].

We argue that the difference between these scores are *not* intrinsic as far as learning network structures is concerned. It has been shown [15] that the entropy score is equivalent to the cross entropy score, and both are equivalent to the CI test. Careful examination of the MDL approach in [7] reveals that it can be analyzed and interpreted using the cross entropy. Analysis of Bayesian score based K2 [1] also shows that its behavior can be described using the language of CI. Such intrinsic equivalence of seemingly different scores should not be too surprising since all these algorithms are trying to learn an approximate minimal I-map which is based on CI by definition.

Second, we consider the graphical representation. Most common algorithms learn a DAG while LIM learns a chordal graph. Pearl [9] shows that an undirected graph cannot represent *induced dependency* but a DAG can. As an example, if x and y are marginally independent causes of z, then the DAG $x \rightarrow z \leftarrow y$ captures all dependence/independence while no undirected graph can. However, DAG is more expressive only if we insist on P-maps. The chordal graph with pairwise connection is the minimal I-map of the above example. Since our concern is learning I-maps as stated in Conjecture 1, the difference between representing outcomes as DAGs and chordal graphs is also nonintrinsic. On the other hand, the use of chordal graphs by LIM *does* simplify our analysis of LIM as will be seen.

Third, LIM only adds type 1 or type 2 links at each step. It seemingly is not paralleled by any common algorithm. We have shown in Theorem 1 that it is a sufficient condition for learning chordal structures. We can also show (not done here due to limit in space) that these links are the *only* links to add that will result in chordal graphs. Hence, the restriction of type 1/2 links is simply the restriction of chordality. Common algorithms that learn BNs must also restrict candidate structures to be acyclic. Given that the difference between DAG and chordal graph representations is not intrinsic (as argued above), the restriction to type 1/2 links by LIM changes nothing to the overall picture.

Finally, common algorithms are able to differentiate between a strong dependence from a weak one in the dataset such that a link corresponding to a weak true dependence or a false dependence due to sampling may be rejected. These algorithms usually select the link to add that corresponds to the strongest dependence among alternatives (indicated by the score) such that the learned structure is as close to the minimal I-map or as sparse as possible. We have chosen to abstract these capabilities out from LIM. Namely, LIM cannot detect a strong dependence from a weak one, cannot reject any noise, nor does LIM try to minimize the links added. It on average will not learn an I-map that is close to minimal and it may sometime (but not always) learn a trivial I-map (complete graph). However, these differences do not affect the truth of Conjecture 1 as minimality is not required.

The important features left in LIM are its single-link lookahead search and its use of a restricted independence test. By using such a simplified algorithm, we hope to demonstrate what is learnable by common algorithms by showing what is learnable by LIM. We hope also to demonstrate models unlearnable by common algorithms, without being distracted by unimportant details of these algorithms. Ultimately, the achievement of these objectives depends on a formal proof of Conjecture 1, which is the goal of our current research. The above positive evidence for Conjecture 1 makes us believe that such a proof is within our reach. Without waiting for such a proof, below we examine the models learnable and unlearnable by LIM.

5 Inadequacy of Common Concepts for Characterization

A characterization of models learnable by LIM should help distinguish models that are learnable and unlearnable by LIM. Can some common concept, e.g., strictly positive models or models with P-maps, be used as such a characterization? In this section, we examine models classifiable using some common concepts and show that these concepts are inadequate to characterize models learnable by LIM.

We briefly review some terms to be used in this section. Two sets of variables X and Y are *marginally independent* if $P(X|Y) = P(X)$, e.g., knowing the value of Y tells us nothing about the value of X. Otherwise, they are marginally dependent. Variables x and y are *logically dependent* if the value of one of them determines uniquely the value of the other. A set N of variables are *generally dependent* if for any proper subset A, $\neg I(A, \phi, N \setminus A)$ holds, i.e., no subset is independent of the rest. A set N of variables are *collectively dependent* if for each proper subset $A \subset N$, there exists no proper subset $C \subset N \setminus A$ such that $P(A|N \setminus A) = P(A|C)$, i.e., no subset can convey all the relevant information between two other subsets.

5.1 Strictly positive models

First, we show that strict positiveness cannot characterize models learnable by LIM.

(x, y, z)	$P(.)$	(x, y, z)	$P(.)$
$(0, 0, 0)$	0.024	$(1, 0, 0)$	0.056
$(0, 0, 1)$	0.216	$(1, 0, 1)$	0.104
$(0, 1, 0)$	0.096	$(1, 1, 0)$	0.024
$(0, 1, 1)$	0.264	$(1, 1, 1)$	0.216

Table 1. A strictly positive model

Example 2 (An unlearnable strictly positive model) Table 1 shows a strictly positive model of three binary variables. The marginals are $P(x = 0) = 0.6$, $P(y = 0) = 0.4$ and $P(z = 0) = 0.2$. Each variable is dependent of the other two, e.g., $P(x|y, z) \neq P(x)$. Therefore, the minimal I-map of the model is a complete graph. However, each pair of variables are marginally independent, e.g., $P(x|y) = P(x)$.

In learning this model, LIM starts with an empty graph. Since the independence test for each of the three type 1 links succeeds, LIM will return the empty graph, which is not an I-map.

Example 2 shows that strict positiveness is not a sufficient condition of learnability by LIM.

Example 3 (A learnable non-positive model) Let M be over $N = \{x, y, z\}$, where x and y are *marginally* dependent, and $z = y$ (*logically* dependent). M is not strictly positive since $P(x, y, z \neq y) = 0$.

For this model, LIM may learn two type 1 links $\{x, y\}$ and $\{y, z\}$, and then halts, which gives a minimal I-map. Alternatively, LIM may learn two type 1 links $\{x, y\}$ and $\{x, z\}$, followed by type 2 link $\{y, z\}$, which gives a trivial I-map.

Example 3 shows that strict positiveness is not a necessary condition of learnability by LIM either.

5.2 Faithful models

A model M that has a P-map is said to be *faithful* [11]. We show that faithfulness does not characterize learnability by LIM.

Example 4 (A learnable unfaithful model) Dead battery and no fuel are two independent causes for a car not to start. Since dead battery and no fuel become dependent given that the car does not start (an induced dependency), the minimal undirected I-map of this model is a complete graph. Since dead battery and no fuel are marginally independent, the I-map is not a D-map. Hence the model is unfaithful.

For this model, LIM will first learn two type 1 links $\{battery, start\}$ and $\{fuel, start\}$. Now $\{battery, fuel\}$ is a type 2 link and the independence test fails. Hence, LIM returns the minimal I-map.

The model in Example 4 is not a graph-isomorph, but it is a DAG isomorph. The next example shows a model that is a graph-isomorph but not a DAG isomorph.

Example 5 (Another learnable unfaithful model) Let M be a graph-isomorph over $N = \{X, Y, Z, W\}$, where the undirected graph has a diamond-shape with links $\{\{X, Y\}, \{Y, Z\}, \{Z, W\}, \{W, X\}\}$. It is not a DAG isomorph [9].

For this model, LIM may first learn any three type 1 links, say, $\{X, Y\}$, $\{Y, Z\}$ and $\{Z, W\}$. It will then learn a type 2 link, say, $\{X, Z\}$, followed by another, $\{W, X\}$. LIM will now halt with the learned graph being a minimal (chordal) I-map.

The next example shows a model that is both a graph-isomorph and a DAG isomorph, but is unlearnable by LIM.

Example 6 (An unlearnable faithful model) Figure 2 (a) shows the P-map of a PDM. For this model, LIM may learn the graph in (b) in the order of link labels. First, four type 1 links are learned, and then three type 2 links are learned. Now, the only type 2 links that may be added are $\{x, z\}$ and $\{y, w\}$. Since the PDM satisfies $I(x, \{y, v\}, z)$ and $I(y, \{z, v\}, w)$, LIM will halt. Note that the link $\{x, w\}$ is not a type 2 link. Note also that since the P-map in (a) is chordal, the PDM is also isomorphic to a DAG.

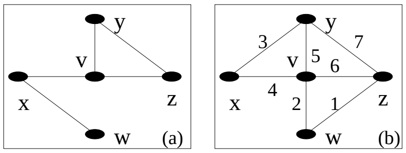

Fig. 2. (a) A P-map of a PDM. (b) A learned structure by LIM.

The above examples show that faithfulness is neither a necessary condition nor a sufficient condition of learnability by LIM, and therefore is not an adequate characterization.

5.3 Pseudo-independent models

A pseudo-independent (PI) model is a probabilistic model where a set of collectively dependent variables displays marginal independence.

Definition 7 *A PDM over a set N of generally dependent variables is* PI *if there exists a partition $\{A_1, \ldots, A_k\}$ $(k > 1)$ of N such that for each $x \in A_i$ and each $y \in A_j$ $(i \neq j)$, x and y are marginally independent.*

Example 2 is a PI model. More elaborated definitions of PI models can be found in [13]. Theorem 8 shows that a necessary condition of learnability by LIM is that M is non-PI.

Theorem 8 *LIM cannot learn an I-map of a PDM M if M is PI.*

Proof:

Let M be a PI model. According to Definition 7, the domain variables can be partitioned into marginally independent subsets. Consider two such subsets A and B. Since LIM starts with an empty graph, A and B are initially disconnected. Since the test "$P(x|y) = P(x)$?" will succeed for each $x \in A$ and each $y \in B$, no type 1 links will ever be added between A and B. Hence, LIM will return a graph G with A and B disconnected.

Since variables in M are generally dependent, any I-map of M must be connected. Hence LIM cannot learn an I-map of M. □

An important relation between PI models and single-link search is the connectivity of the learned graph shown by Theorem 9.

Theorem 9 *Let M be a generally dependent PDM over N and G be a graph learned by LIM from M. Then G is connected iff M is non-PI.*

Proof:

The necessity is clear from the proof of Theorem 8. We show the sufficiency below:

Assume that M is generally dependent and non-PI. Then there is no marginally independent partitions of N. Hence LIM will be able to find type 1 links which fail the independence test until the learned graph is connected. □

PI models are not the only type of models unlearnable by LIM. The following example demonstrates this.

Example 10 (An unlearnable non-PI model) Table 2 shows a model of four variables. It is non-PI since $\neg I(x, \phi, y)$, $\neg I(y, \phi, z)$ and $\neg I(z, \phi, w)$.

Its minimal I-map is a complete graph, which can be inferred as follows: If there exists a minimal I-map that is not complete, then at least one pair of variables is not connected directly. This pair must then be independent given the other two. However for each variable, its distribution conditioned on the other three variables is *not* degenerated. For example, no conditioning variable may be removed in $P(x|y, z, w)$ without changing the distribution.

In learning this model, LIM may first learn three type 1 links $\{x, y\}$, $\{y, z\}$ and $\{z, w\}$. Now only two type 2 links $\{x, z\}$ and $\{y, w\}$ may be added. However, since this model satisfies $I(x, y, z)$ and $I(y, z, w)$, both type 2 links will be rejected. Hence, LIM will return a graph with only the three type 1 links, which is not an I-map.

(x, y, z, w)	$P(.)$	(x, y, z, w)	$P(.)$	(x, y, z, w)	$P(.)$	(x, y, z, w)	$P(.)$
$(0, 0, 0, 0)$	0.4192	$(0, 1, 0, 0)$	0.0189	$(1, 0, 0, 0)$	0.0548	$(1, 1, 0, 0)$	0.0613
$(0, 0, 0, 1)$	0.0725	$(0, 1, 0, 1)$	0.0005	$(1, 0, 0, 1)$	0.0088	$(1, 1, 0, 1)$	0.0132
$(0, 0, 1, 0)$	0.0690	$(0, 1, 1, 0)$	0.0065	$(1, 0, 1, 0)$	0.0156	$(1, 1, 1, 0)$	0.0773
$(0, 0, 1, 1)$	0.0871	$(0, 1, 1, 1)$	0.0296	$(1, 0, 1, 1)$	0.0045	$(1, 1, 1, 1)$	0.0611

Table 2. A non-PI model

Example 10 shows that although being a non-PI model is a necessary condition for learnability by LIM, it is not a sufficient condition. Hence pseudo-independence cannot characterize models learnable by LIM.

Example 10 is in fact a positive and non-PI model. Therefore, it also shows that the combination of positiveness and non-PI is still not a sufficient condition for learnability by LIM.

5.4 On the effect of greedy search

For Example 10, one might wonder if a greedy search may change the situation. That is *not* the case. Among the six potential links, the three links learned above have stronger dependence between their endpoints, measured by *average mutual information*, compared with the other three links. Therefore, even if LIM is augmented with the ability to compare the strength of dependence among alternative links and modified into a greedy search algorithm, the learning outcome will still be the same as described in Example 10.

On the other hand, if LIM chooses an order different from a greedy search, it may be able to learn the I-map of the model in Example 10. For example, it may first learn type 1 links $\{x, y\}$, $\{x, z\}$ and $\{z, w\}$. The type 2 link $\{y, z\}$ and then $\{x, w\}$ can then be learned since the corresponding independence tests will fail. Finally, the type 2 link $\{y, w\}$ will be learned.

Note that we are not suggesting learning a complete graph in general. The above example can be easily extended into a sparse model with more variables while keeping the dependence among $\{x, y, z, w\}$ unchanged. Hence, the example only illustrates a subprocess in learning a generally much large model.

6 Characterization of LIM-learnable Models

In this section, we show that the class of PDMs learnable by LIM can be characterized by the following properties:

Definition 11 *Let X, Y, Z, V and W be any disjoint subsets of variables.*

Composition: $I(X, Y, Z)$ & $I(X, Y, W) \Longrightarrow I(X, Y, Z \cup W)$.
Strong Transitivity: $I(X, Y \cup V, Z)$ & $I(Y, Z \cup V, W) \Longrightarrow I(X, Y \cup V, Z \cup W)$.

We will place proofs for some formal results in Appendix for readability.

We shall consider only chordal graphs as candidate I-maps of PDMs. We shall use a junction tree (JT) of a chordal graph in our investigation. A JT T of a chordal graph G is a tree. Each node in T is labeled by a (maximal) clique of G and each link, called a *sepset*, is labeled by the intersection of the two cliques at its ends. T is so connected that the intersection of any two cliques is contained in each sepset on the unique path between them.

To ensure the validity of any conclusion drawn from the JT, we need to establish the equivalence of a chordal graph and its JTs as I-maps. We complete a partial result by Pearl in Theorem 13 below.

Conditional independence is portrayed in an I-map by graphical separation. We define graphical separation in a JT of a chordal graph as follows:

Definition 12 *Let T be a JT of cliques of a chordal graph G. For any disjoint subsets X, Y and Z of nodes in G, X and Y are* `s-separated` *by Z in T, denoted by $< X|Z|Y >_T$, if for each $x \in X$, $y \in Y$ and each two cliques C_x, C_y in T such that $x \in C_x$ and $y \in C_y$,*

1. *$C_x \neq C_y$, and*
2. *on the path between C_x and C_y in T, there is a sepset $S \subseteq Z$.*

The following theorem shows that, using s-separation, a JT of a chordal graph portrays exactly the same set of relations of graphical separation as its deriving chordal graph. The sufficiency has been shown in [9]. We prove here the necessity.

Theorem 13 *Let T be a JT of cliques for a connected chordal graph G. For any disjoint subsets X, Y and Z of nodes in G, $< X|Z|Y >_G \iff < X|Z|Y >_T$.*

Next, we show that for any PDM that satisfies Composition and Strong Transitivity, the dependence structure learned by LIM will be an I-map. Due to the equivalence of a chordal graph G and its JT T as I-maps (Theorem 13), we need only to show that $< X|Z|Y >_T \implies I(X, Z, Y)$ holds for any G learned by LIM.

In the following formal results, we sometime assume a *generally dependent* PDM. This is not a restriction of the learnable models but rather a simplification of proofs. When the underlying PDM is not generally dependent, our result is applicable to each independent submodel.

Theorem 14 shows that the Composition axiom rules out PI models. It is also needed by Lemma 15.

Theorem 14 *Let M be a generally dependent PDM over N that satisfies Composition. Then M is non-PI.*

Lemma 15 shows that if a PDM satisfies Composition and Strong Transitivity, then in any graph learned by LIM, a clique sepset portrays conditional independence correctly.

Lemma 15 *Let M be a generally dependent PDM over N that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM and T be a JT of G.*

Then $I(C_a \setminus S, S, C_b \setminus S)$ holds for each pair of cliques C_a and C_b in T where S is a sepset on the path between C_a and C_b.

Lemma 16 extends Lemma 15 by allowing the separating subset to be any superset of a clique sepset.

Lemma 16 *Let M be a generally dependent PDM over N that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM and T be a JT of G.*

Then $I(C_a \setminus Q, Q, C_b \setminus Q)$ holds for each pair of cliques C_a and C_b in T where Q contains a sepset on the path between C_a and C_b.

Finally, we extend Lemma 16 to conditional independence of any subsets.

Theorem 17 *Let M be a generally dependent PDM over N that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM and T be a JT of G. Let X, Y, Z be any disjoint subsets of N such that $< X|Z|Y >_T$ holds according to s-separation.*

Then $I(X, Z, Y)$ holds.

Proof: Let $X_1, ..., X_m$ be all cliques in T such that $X \cap X_i \neq \phi$ $(1 \leq i \leq m)$, and $Y_1, ..., Y_n$ be all cliques in T such that $Y \cap Y_j \neq \phi$ $(1 \leq j \leq n)$. For each X_i and each Y_j, we have $I(X_i, Z, Y_j)$ by Lemma 16. Applying Composition to Y_j $(1 \leq j \leq n)$, we have $I(X_i, Z, Y)$ for each given i. Applying Composition to X_i $(1 \leq i \leq m)$, we obtain $I(X, Z, Y)$. □

Theorem 17, together with Theorem 13, implies that LIM will return an I-map as long as the underlying PDM satisfies Composition and Strong Transitivity. This is summarized in Corollary 18. Note that the general dependence can now be removed.

Corollary 18 *Let M be a PDM that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM. Then G is an I-map of M.*

7 Remarks

Corollary 18 provides an axiomatic characterization of models learnable by LIM. Coupled with Conjecture 1, it implies that a PDM satisfying Composition and Strong Transitivity is learnable by *any* algorithm, for learning BNs or DMNs, equipped with a single-link search and some scoring metric equivalent to a conditional independence test.

Can PDMs violating Composition be learned by LIM in general? Theorems 14 and 9 show PI models as PDMs that violate Composition and are unlearnable by LIM. Conjecture 1 then implies that PI models are unlearnable by *any* algorithm, for learning BNs or DMNs, equipped with a single-link search and some scoring metric equivalent to a conditional independence test.

Can PDMs violating Strong Transitivity be learned by LIM in general? Example 10 shows the kind of non-PI PDMs that violate Strong Transitivity and are not learnable by LIM when certain search paths (including greedy search) are followed. Conjecture 1 then implies that if Strong Transitivity does not hold in a PDM, the learning outcome is likely to be incorrect for *any* belief network learning algorithm, equipped with a single-link search and some scoring metric equivalent to a conditional independence test, and followed a *single* search path.

Our characterization of learnability by LIM may be compared with faithfulness as follows: Both graph-isomorph and DAG isomorph are closely tied to strict positiveness through the Intersection axiom (Section 2). Our characterization of learnability by LIM does not require Intersection and therefore does not depend on strict positiveness. This can be seen from Example 3 which violates Intersection but is learnable. DAG isomorph also requires the Chordality axiom. It is not required by our characterization as can also be seen from Example 5. Hence, LIM-learnable models are *not* a subset of either graph-isomorph or DAG isomorph. In other words, LIM-learnable models are *not* a subset of faithful models.

On the other hand, our characterization requires Strong Transitivity, which differs from the Transitivity axiom for graph-isomorph and the Weak Transitivity for DAG isomorph. As shown in Example 6, a PDM that is both a graph-isomorph and a DAG isomorph can violate Strong Transitivity. Hence, faithful models are *not* a subset of models characterized by Composition and Strong Transitivity.

These results improve our understanding of common algorithms for learning belief networks, and suggest useful directions for improving these algorithms. We are currently working on a formal proof of Conjecture 1 to put these results in firm ground.

Acknowledgement

This work is supported by grant OGP0155425 from NSERC, a grant from IRIS in the Networks of Centres of Excellence Program of Canada, a grant from SaskTel and a SaskTel CRD grant from NSERC. Helpful comments were received from Y. Huang.

Appendix: Proofs

Theorem 13 Let T be a JT of cliques for a connected chordal graph G. For any disjoint subsets X, Y and Z of nodes in G, $< X|Z|Y >_G \iff < X|Z|Y >_T$.
Proof:

The proof for $< X|Z|Y >_G \implies < X|Z|Y >_T$ can be found in [9] (Lemma 1, p114). We show $< X|Z|Y >_G \impliedby < X|Z|Y >_T$.

Let $x \in X$ and $y \in Y$ be contained in cliques C_x and C_y in T, respectively. Suppose that on the path between C_x and C_y, there is a sepset $S \subseteq Z$. We show that $< x|S|y >_G$ holds and so does $< x|Z|y >_G$.

Assume that $< x|S|y >_G$ does not hold. Then there exists a path $(x, v_1, v_2, ..., v_n, y)$ in G not through S. That is, $v_i \notin S$ for $1 \leq i \leq n$.

On the other hand, if v_{i-1}, v_i and v_{i+1} are not contained in a same clique in T such that v_{i-1}, v_i are in clique C_{i-1} and v_i, v_{i+1} are in C_{i+1}, then on the unique path between C_{i-1} and C_{i+1} in T, every sepset must contain v_i. To accommodate the case where $n = 1$, we shall denote $v_0 = x$ and $v_{n+1} = y$. Hence on the path from C_x to C_y in T, every sepset contains at least one $v_i \notin S$. This contradicts that S is a sepset between C_x and C_y. □

Theorem 14 Let M be a generally dependent PDM over N that satisfies Composition. Then M is non-PI.
Proof:

We shall build a subset S of N from a singleton such that each new element of S is dependent on at least one existing element of S. When $S = N$, we have shown that M is non-PI. We prove by induction on the cardinality of S.

Let $S_1 = \{x\}$ for any $x \in N$. We search for $y \in N \setminus \{x\}$ such that $\neg I(x, \phi, y)$. If $I(x, \phi, y_1)$ holds for $y_1 \in N \setminus \{x\}$, then from general dependence and Composition (contrapositive form) of M, we have

$$\neg I(x, \phi, N \setminus \{x\}) \ \ \& \ \ I(x, \phi, y_1) \Longrightarrow \neg I(x, \phi, N \setminus \{x, y_1\}).$$

We then search for $y \in N \setminus \{x, y_1\}$ such that $\neg I(x, \phi, y)$. By recursively applying general dependence over the remaining subset and Composition, we will find $y \in N \setminus \{x\}$ such that $\neg I(x, \phi, y)$. We update $S_2 = \{x, y\}$.

Suppose we have updated S_i $(i > 1)$. Next we search for $v \in S_i$ and $z \in N \setminus S_i$ such that $\neg I(v, \phi, z)$ holds. If $I(v_1, \phi, z)$ holds for $v_1 \in S_i$ and each $z \in N \setminus S_i$, then by Composition we have $I(v_1, \phi, N \setminus S_i)$. In that case, by general dependence and Composition of M we have

$$\neg I(N \setminus S_i, \phi, S_i) \ \ \& \ \ I(N \setminus S_i, \phi, v_1) \Longrightarrow \neg I(N \setminus S_i, \phi, S_i \setminus \{v_1\}).$$

If $I(v_2, \phi, z)$ holds for $v_2 \in S_i \setminus \{v_1\}$ and each $z \in N \setminus S_i$, then by Composition we have $I(v_2, \phi, N \setminus S_i)$. In that case, by general dependence and Composition of M we have

$$\neg I(N \setminus S_i, \phi, S_i) \ \ \& \ \ I(N \setminus S_i, \phi, \{v_1, v_2\}) \Longrightarrow \neg I(N \setminus S_i, \phi, S_i \setminus \{v_1, v_2\}).$$

Repeating this argument, we eventually will find $v \in S_i$ and $z \in N \setminus S_i$ such that $\neg I(v, \phi, z)$ holds. We can then update $S_{i+1} = S_i \cup \{z\}$. □

Lemma 15 Let M be a generally dependent PDM over N that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM and T be a JT of G.

Then $I(C_a \setminus S, S, C_b \setminus S)$ holds for each pair of cliques C_a and C_b in T where S is a sepset on the path between C_a and C_b.
Proof:

By Theorem 14, M is non-PI. By Theorems 1 and 9, G is chordal and connected. Hence T exists.

We first show that $I(C_a \setminus C_b, C_a \cap C_b, C_b \setminus C_a)$ holds for any adjacent C_a and C_b. LIM halts only if $I(x, C_a \cap C_b, y)$ holds for each pair of adjacent cliques C_a, C_b in T and each pair of nodes $x \in C_a \setminus C_b$ and $y \in C_b \setminus C_a$. Otherwise, $\{x, y\}$ is a type 2 link that fails the independence test. Assume that $I(x, C_a \cap C_b, Y)$ holds, where $Y \subset C_b \setminus C_a$. Let $Y' = Y \cup \{y'\}$ where $y' \in C_b \setminus (C_a \cup Y)$. From Composition,

$$I(x, C_a \cap C_b, y') \ \ \& \ \ I(x, C_a \cap C_b, Y) \implies I(x, C_a \cap C_b, Y').$$

Hence, for each $x \in C_a \setminus C_b$, we have $I(x, C_a \cap C_b, C_b \setminus C_a)$. Assume that $I(X, C_a \cap C_b, C_b \setminus C_a)$ holds, where $X \subset C_a \setminus C_b$. Let $X' = X \cup \{x'\}$ where $x' \in C_a \setminus (C_b \cup X)$. From Composition,

$$I(X, C_a \cap C_b, C_b \setminus C_a) \ \ \& \ \ I(x', C_a \cap C_b, C_b \setminus C_a) \implies I(X', C_a \cap C_b, C_b \setminus C_a).$$

Hence, $I(C_a \setminus C_b, C_a \cap C_b, C_b \setminus C_a)$ holds for each pair of adjacent C_a and C_b.

Next, we show that $I(C_a \setminus S, S, C_b \setminus S)$ holds for non-adjacent C_a and C_b, where S is a sepset on the path between C_a and C_b. Let three adjacent cliques $C_a = X \cup Y \cup V$, $C_1 = Y \cup V \cup Z \cup A$ and $C_b = V \cup Z \cup W$ form a chain $C_a - C_1 - C_b$ in T, where each letter denotes a disjoint subset of variables. We show that $I(X, Y \cup V, Z \cup W)$ holds. From the above proof and Decomposition, we have $I(X, Y \cup V, Z)$ and $I(Y, V \cup Z, W)$. From Strong Transitivity, we conclude $I(X, Y \cup V, Z \cup W)$.

Now consider a chain of $n \geq 1$ intermediate cliques in T, $C_a - C_1 - ... - C_n - C_b$. We denote $C_i \setminus \cup_{j \neq i} C_j$ by R_i and denote $C_i \setminus R_i$ by D_i for $1 \leq i \leq n$. R_i is the subset of C_i not contained in any other cliques. It is irrelevant here as can be seen from the subset A above. Note $(D_{i-1} \cap D_i) \cup (D_i \cap D_{i+1}) = D_i$ but $(C_{i-1} \cap C_i) \cup (C_i \cap C_{i+1}) = C_i$ is not true in general.

Assume $I(C_a \setminus S, S, C_b \setminus S)$ holds for $n = m \geq 1$. When $n = m + 1$, the clique chain becomes $C_a - C_1 - ... - C_m - C_n - C_b$. We show that $I(C_a \setminus S, S, C_b \setminus S)$ still holds where S is a sepset on the chain.

Let S be the sepset between C_{i-1} and C_i ($i \leq n$), and S' be the sepset between C_i and C_{i+1}. Note that S is a sepset in the subchain from C_a to C_i, and S' is a sepset in the subchain from C_i to C_b. Either subchain has no more than m intermediate cliques. From the assumption and Decomposition, we have

$$I(C_a \setminus S, S, D_i \setminus S) \quad and \quad I(D_i \setminus S', S', C_b \setminus S').$$

Since $D_i = S \cup S'$, we have $S \supseteq D_i \setminus S'$ and $S' \supseteq D_i \setminus S$. From Strong Transitivity with

$$X = C_a \setminus S, \quad Y = D_i \setminus S', \quad Z = D_i \setminus S, \quad V = S \cap S', \quad and \quad W = C_b \setminus S',$$

we conclude $I(C_a \setminus S, S, (D_i \setminus S) \cup (C_b \setminus S'))$.

We now only have to show $(D_i \setminus S) \cup (C_b \setminus S') \supseteq C_b \setminus S$.

Since $D_i \setminus S = S' \setminus S$, and $S' = (S' \setminus S) \cup (S' \cap S)$, we have

$$(D_i \setminus S) \cup (C_b \setminus S') = (S' \setminus S) \cup (C_b \setminus ((S' \setminus S) \cup (S' \cap S))) = (S' \setminus S) \cup (C_b \setminus (S \cap S')).$$

Since $C_b \cap (S \setminus S') = \phi$ (T is a JT), we obtain

$$C_b \setminus (S \cap S') = C_b \setminus ((S' \cap S) \cup (S \setminus S')) = C_b \setminus S.$$

Hence, $(D_i \setminus S) \cup (C_b \setminus S') = (S' \setminus S) \cup (C_b \setminus S) \supseteq C_b \setminus S$. □

Lemma 16 Let M be a generally dependent PDM over N that satisfies Composition and Strong Transitivity. Let G be a chordal graph returned by LIM and T be a JT of G.

Then $I(C_a \setminus Q, Q, C_b \setminus Q)$ holds for each pair of cliques C_a and C_b in T where Q contains a sepset on the path between C_a and C_b.

Proof:

Let the sepset between C_a and C_b be $S \subseteq Q$. We have $I(C_a \setminus Q, S, C_b \setminus Q)$ by Lemma 15 and Decomposition. Given S, T is partitioned into two subtrees T_a (containing C_a) and T_b (containing C_b). Let Q_a (Q_b) be the subset of $Q \setminus S$ that is contained in T_a (T_b).

For each variable $y \in Q_b$, we have $I(C_a \setminus Q, S, y)$ by Lemma 15. By Composition, we derive $I(C_a \setminus Q, S, Q_b \cup (C_b \setminus Q))$. From Weak Union, we have $I(C_a \setminus Q, S \cup Q_b, C_b \setminus Q)$. For each variable $x \in Q_a$, from the symmetry of x and $C_a \setminus Q$, we have $I(x, S \cup Q_b, C_b \setminus Q)$. By Composition, we derive $I(Q_a \cup (C_a \setminus Q), S \cup Q_b, C_b \setminus Q)$. From Weak Union, we have $I(C_a \setminus Q, S \cup Q_b \cup Q_a, C_b \setminus Q) = I(C_a \setminus Q, Q, C_b \setminus Q)$. □

References

1. G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, (9):309–347, 1992.
2. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
3. E.H. Herskovits and G.F. Cooper. Kutato: an entropy-driven system for construction of probabilistic expert systems from database. In *Proc. 6th Conf. on Uncertainty in Artificial Intelligence*, pages 54–62, Cambridge,, 1990.
4. J. Hu and Y. Xiang. Learning belief networks in domains with recursively embedded pseudo independent submodels. In *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, pages 258–265, Providence, 1997.
5. F.V. Jensen. *An introduction to Bayesian networks.* UCL Press, 1996.
6. F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, (4):269–282, 1990.
7. W. Lam and F. Bacchus. Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.
8. S.L. Lauritzen, A.P. Dawid, B.N. Larsen, and H.G. Leimer. Independence properties of directed Markov fields. *Networks*, 20:491–505, 1990.
9. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.
10. G. Rebane and J. Pearl. The recovery of causal ploy-trees from statistical data. In *Proc. of Workshop on Uncertainty in Artificial Intelligence*, pages 222–228, Seattle, 1987.
11. P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and Search.* Springer-Verlag, 1993.
12. S.K.M. Wong and Y. Xiang. Construction of a Markov network from data for probabilistic inference. In *Proc. 3rd Inter. Workshop on Rough Sets and Soft Computing*, pages 562–569, San Jose, 1994.
13. Y. Xiang. Towards understanding of pseudo-independent domains. In *Poster Proc. 10th Inter. Symposium on Methodologies for Intelligent Systems*, Oct 1997.

14. Y. Xiang, D. Poole, and M. P. Beddoes. Multiply sectioned Bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence*, 9(2):171–220, 1993.
15. Y. Xiang, S.K.M. Wong, and N. Cercone. A 'microscopic' study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26(1):65–92, 1997.

This article was processed using the LaTeX macro package with LLNCS style