Monotone DAG Faithfulness: A Bad Assumption

David Maxwell Chickering   Christopher Meek

Technical Report
MSR-TR-2003-16

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

In a recent paper, Cheng, Greiner, Kelly, Bell and Liu (Artificial Intelligence 137:43-90; 2002) describe an algorithm for learning Bayesian networks that—in a domain consisting of n variables—identifies the optimal solution using $O(n^4)$ calls to a mutual-information oracle. This seemingly incredible result relies on (1) the standard assumption that the generative distribution is Markov and faithful to some directed acyclic graph (DAG), and (2) a new assumption about the generative distribution that the authors call *monotone DAG faithfulness (MDF)*. The MDF assumption rests on an intuitive connection between active paths in a Bayesian-network structure and the mutual information among variables. The assumption states that the (conditional) mutual information between a pair of variables is a monotonic function of the set of active paths between those variables; the more active paths between the variables the higher the mutual information. In this paper, we demonstrate the unfortunate result that, for any realistic learning scenario, the monotone DAG faithfulness assumption is *incompatible* with the faithfulness assumption. In fact, by assuming both MDF and faithfulness, we restrict the class of possible Bayesian-network structures to one for which the optimal solution can be identified with $\mathrm{O}(n^2)$ calls to an independence oracle.

1 Introduction

Learning Bayesian networks from data has traditionally been considered a hard problem by most researchers. Numerous papers have demonstrated that, under a number of different scenarios, identifying the "best" Bayesian-network structure is NP-hard. In a recent paper describing information-theoretic approaches to this learning problem, however, Cheng, Greiner, Kelly, Bell and Liu (2002) (hereafter CGKBL) describe an algorithm that runs in polynomial time when given a mutual-information oracle. In particular, for a domain of n variables, CGKBL claim that the algorithm identifies the generative Bayesian-network structure using $\mathrm{O}(n^4)$ calls to the oracle, regardless of the complexity of that generative network. The seemingly incredible result relies on an assumption about the generative distribution that CGKBL call *monotone DAG faithfulness.* Intuitively, the assumption states that in a distribution that is perfect with respect to some Bayesian-network structure $\mathcal{G}$, the (conditional) mutual information between two variables is a monotonic function of the "active paths" between those variables in $\mathcal{G}$.

In this paper, we demonstrate that monotone DAG faithfulness is a bad assumption. In particular, we show that the standard faithfulness assumption and the monotone DAG faithfulness assumption are inconsistent with each other unless we restrict the possible generative structures to an unreasonably simple class; furthermore, the optimal member of this simple class of models can be identified using a standard independence-based learning algorithm using only $\mathrm{O}(n^2)$ calls to an independence oracle. Unfortunately, our results cast doubt once again on the existence of an efficient and correct Bayesian-network learning algorithm under reasonable assumptions.

The paper is organized as follows. In Section 2, we provide background material and define the monotone DAG faithfulness assumption more rigorously. In Section 3, we describe a family of independence-based and information-based learning algorithms, consider the worst-case complexity of these algorithms, and show how the monotone DAG faithfulness assumption can lead to the incredible result of CGKBL. In Section 4, we provide simple examples that highlight the problems with the monotone DAG faithfulness assumption, and we prove that the assumption is incompatible with faithfulness unless we impose severe restrictions on the generative structure. Finally, in Section 5, we conclude with a discussion.

2 Background

In this section, we describe our notation and present relevant background material. We assume that the reader has some basic familiarity with probability theory, graph theory, and Bayesian networks.

A *Bayesian network*, which is used to represent a joint distribution over the variables in a domain, consists of (1) a directed acyclic graph (or *DAG* for short) in which there is a single vertex associated with each variable in the domain, and (2) a corresponding set of parameters that defines the joint distribution. We use the calligraphic letter $\mathcal{G}$ to denote a Bayesian-network structure. We use *variable* to denote both a random variable in the domain and the corresponding vertex (or node) in the Bayesian-network structure. Thus, for example, we might say that variable X is adjacent to variable Y in Bayesian-network structure $\mathcal{G}$. The parameters of a Bayesian network specify the conditional distribution of each variable given its parents in the graph, and the joint distribution for the variables in the domain is defined by the product of these conditional distributions. For more information see, for example, Pearl (1988).

We use bold-faced Roman letters for sets of variables (e.g., $\mathbf{X}$), non-bold-faced Roman letters for singleton variables (e.g., X) and lower-case Roman letters for values of the variables (e.g., $\mathbf{X} = \mathbf{x}$, $X = x$). To simplify notation when expressing probabilities, we omit the name of the variables involved. For example, we use $p(y|\mathbf{x})$ instead of $p(Y = y|\mathbf{X} = \mathbf{x})$. For a distribution p, we use $Ind_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ to denote the fact that in p, $\mathbf{X}$ is independent of $\mathbf{Y}$ given conditioning set $\mathbf{Z}$. When the conditioning set is empty, we use $Ind_p(\mathbf{X}; \mathbf{Y})$ instead. To simplify notation, we omit the standard set notation when considering a singleton variable in any position. For example, we use $Ind_p(X; Y|\mathbf{Z})$ instead of $Ind_p(\{X\}; \{Y\}|\mathbf{Z})$.

2.1 Independence Constraints of DAGs

Any joint distribution represented by a Bayesian network must satisfy certain independence constraints that are imposed by the structure of the model. Because a Bayesian network represents a joint distribution as the product of conditional distributions, the joint distribution must satisfy the *Markov conditions* of the structure: each variable must be independent of its non-descendants given its parents. The Markov conditions constitute a basis for the independence facts that are true for all distributions that can be represented by a Bayesian network with a given structure. The *d-separation* criterion is a graphical criterion that characterizes all of these *structural* independence constraints. In order to define the criterion, we first need to define an *active path*. We provide two distinct definitions for an active path, both of which are adequate for defining the d-separation criterion. Both definitions are standard, and we include both to highlight the sensitivity of the MDF assumption to the definition.

Before proceeding, we provide standard definitions for a path, a simple path, and a collider. A *path* π in a graph $\mathcal{G}$ is an ordered sequence of variables $(X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ such that for each $\{X_{(i)}, X_{(i+1)}\}$, either the edge $X_{(i)} \rightarrow X_{(i+1)}$ or the edge $X_{(i)} \leftarrow X_{(i+1)}$ exists in $\mathcal{G}$, where $X_{(i)}$ denotes the variable at position i on the path. A path is a *simple* *path* if each variable occurs at most once in the path. Three (ordered) variables (X, Y, Z) form a *collider complex* in $\mathcal{G}$ if the edges $X \rightarrow Y$ and $Y \leftarrow Z$ are both contained in $\mathcal{G}$. A variable $X_{(i)}$ is a *collider* at position i in a path $\pi = (X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ in graph $\mathcal{G}$ if $1 < i < n$ and $(X_{(i-1)}, X_{(i)}, X_{(i+1)})$ is a collider complex in $\mathcal{G}$. Note that a collider is defined by not only a variable, but the *position* of that variable in a path; a particular variable may appear both as a collider and as a non-collider within a path.

We now provide our two definitions:

Definition 1 (Compound Active Path) *A path $\pi = (X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ is a* compound active path *given conditioning set $\mathbf{Y}$ in DAG $\mathcal{G}$ if each variable $X_{(i)}$ in the path has one of the two following properties: (1) $X_{(i)}$ is not a collider at position i and $X_{(i)}$ is not in $\mathbf{Y}$, or (2) $X_{(i)}$ is a collider at position i and either $X_{(i)}$ or a descendant of $X_{(i)}$ in $\mathcal{G}$ is in $\mathbf{Y}$.*

Definition 2 (Simple Active Path) *A path π is a* simple active path *given conditioning set $\mathbf{Y}$ in DAG $\mathcal{G}$ if π is a compound active path given $\mathbf{Y}$ in $\mathcal{G}$ that is simple.*

Note that the endpoints of a path cannot be colliders. This means that under either definition of an active path, the endpoints cannot be in the conditioning set. To emphasize the distinction between the two definitions above, consider the graph in Figure 1. Given conditioning set D, there is exactly one simple active path between A and B, namely, $A \rightarrow C \leftarrow B$. Given this same conditioning set, there are additional compound active paths including $A \rightarrow C \rightarrow D \leftarrow C \leftarrow B$. In fact, there are an infinite number of these additional paths as we can, for example, prepend $A \rightarrow C \leftarrow A$ to any compound active path and the result is a compound active path.

The following proposition, which is proved in the appendix, establishes a relationship between simple and compound active paths that is important for the definition of d-separation.

Proposition 1 *There is a simple active path between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$ if and only if there is a compound active path between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$.*

Finally, we can define the d-separation criterion. Sets of variables $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated* given a set of variables $\mathbf{Z}$ in $\mathcal{G}$ if there does not exist a simple active path between a variable in $\mathbf{X}$ and a variable in $\mathbf{Y}$ given conditioning set $\mathbf{Z}$. From Proposition 1, we see that d-separation is equivalently defined by the absence of a compound active path. We use $Dsep_{\mathcal{G}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ to denote that $\mathbf{X}$ is d-separated from $\mathbf{Y}$ given $\mathbf{Z}$ in $\mathcal{G}$.

The d-separation criterion provides a useful connection between a DAG and the corresponding set distributions that can be represented with a Bayesian network with that structure. In particular, Pearl (1988) shows that if $Dsep_{\mathcal{G}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$, then for any distribution p that can be represented by a Bayesian network with structure $\mathcal{G}$, it must be the case that $Ind_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$.[1] Given this strong connection between d-separation and representability in a Bayesian network, it is natural to define the following property for a distribution.

[1] This is the *soundness* result for d-separation. Pearl (1988) also shows that d-separation is *complete*; that is, if $Ind_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ for every p that can be represented by a Bayesian network with structure $\mathcal{G}$, then $Dsep_{\mathcal{G}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$.

Definition 3 (Markov Distribution) *A distribution p is* Markov with respect to $\mathcal{G}$ *if $Dsep_{\mathcal{G}}(\mathbf{X};\mathbf{Y}|\mathbf{Z})$ implies $Ind_p(\mathbf{X};\mathbf{Y}|\mathbf{Z})$.*

We use $\mathbf{Markov}(\mathcal{G})$ to denote the set of distributions that are Markov with respect to $\mathcal{G}$.

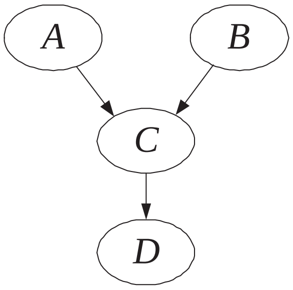

Figure 1: A simple Bayesian-network structure.

If two DAGs $\mathcal{G}$ and $\mathcal{G}'$ represent the same independence constraints, we say that they are *equivalent*. Verma and Pearl (1991) shows that two DAGs are equivalent if and only if (1) they have the same adjacencies and (2) for any collider complex (X, Y, Z) in one of the DAGs such that X and Z are not adjacent, this "v-structure" also exists in the other DAG.

2.2 Faithfulness

The Markov property provides a connection between the structure of a Bayesian network and independence. Namely, the absence of an edge guarantees a set of independence facts. The existence of an edge between variable X and Y in the structure $\mathcal{G}$, however, does not guarantee that a Bayesian network with structure $\mathcal{G}$ will exhibit a dependence between X and Y. Without making assumptions connecting the existence of edges in a generative structure and the joint distribution of a generative Bayesian network, it is not generally possible to recover the generative Bayesian-network structure.

Most structure-learning algorithms that have large-sample correctness guarantees assume that the distribution from which the data is generated is both Markov and *faithful* with respect to some DAG. Functionally, the faithfulness assumption means that every edge in this DAG can be identified by a *lack* of independence in the generative distribution, for every conditioning set, between the corresponding endpoint variables.

Definition 4 (Faithful Distribution) *A distribution p is* faithful *to $\mathcal{G}$ if $Ind_p(\mathbf{X};\mathbf{Y}|\mathbf{Z})$ implies $Dsep_{\mathcal{G}}(\mathbf{X};\mathbf{Y}|\mathbf{Z})$.*

We use $\mathbf{Faithful}(\mathcal{G})$ to denote the set of distributions that are faithful to $\mathcal{G}$. As we see in the next section, the intersection $\mathbf{Markov}(\mathcal{G}) \cap \mathbf{Faithful}(\mathcal{G})$ is an important class of distributions for proving optimality results about learning algorithms; we use $\mathbf{Perfect}(\mathcal{G})$ to denote this intersection. For a distribution p, if there exists a DAG $\mathcal{G}$ such that $p \in \mathbf{Perfect}(\mathcal{G})$, we say that p is a *DAG-perfect* distribution, and that *p is perfect with respect to $\mathcal{G}$.*

The assumption of faithfulness might seem like an unjustifiably strong assumption, but a joint distribution represented by a Bayesian network can fail to be faithful only by a precise balancing of the parameters. This intuition is made more precise in Meek (1995) and Spirtes, Glymour, and Scheines (2000) where it is shown that almost all distributions that are Markov with respect to a structure $\mathcal{G}$ are also faithful to that structure. In other words, if you put a smooth measure over the distributions representable by a Bayesian network with structure $\mathcal{G}$ and choose a distribution at random, you will choose a faithful distribution with probability one.

2.3 Information and Monotone DAG Faithfulness

The CGKBL algorithm uses the conditional-mutual information between sets of variables to recover the structure of a Bayesian network. The correctness claims of CGKBL are based on an assumption that they call *monotone DAG faithfulness.* Similar to the assumption of faithfulness, this assumption connects properties of the generative Bayesian-network structure and the information relationships among sets of variables in the generative distribution.

The conditional mutual information between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ is formally defined as:

$$Inf_p(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = \sum_{\mathbf{x},\mathbf{y},\mathbf{z}} p(\mathbf{x},\mathbf{y},\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{y}|\mathbf{z})}{p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})}. \tag{1}$$

where ‘log’ denotes the base-two logarithm.

In the previous section we defined two types of active paths: simple active paths and compound active paths. Active paths as defined by CGKBL are compound active paths. We include the alternative simple definition because it is a standard definition of active path and because it highlights the sensitivity of the monotone DAG faithfulness assumption to the underlying definition of active path. We find it convenient to refer to an *active path* in a DAG with the understanding that we have a specific definition—either a simple active path or a compound active path—in mind.

We now provide a formal definition of monotone DAG faithfulness (MDF). Let $Active^s_{\mathcal{G}}(X;Y|\mathbf{Z})$ denote the set of simple active paths between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$. Similarly, let $Active^c_{\mathcal{G}}(X;Y|\mathbf{Z})$ denote the set of compound active paths between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$. We use $Active_{\mathcal{G}}(X;Y|\mathbf{Z})$ to denote the set of active paths under one of the two definitions of active path when we want to avoid specifying which definition of active path to use.

Definition 5 (Simple Monotone DAG Faithfulness) *A distribution p is* simple monotone DAG faithful *with respect to a DAG $\mathcal{G}$ if*

$$Active^s_{\mathcal{G}}(X;Y|\mathbf{Z}) \subseteq Active^s_{\mathcal{G}}(X;Y|\mathbf{Z}') \Rightarrow Inf_p(X;Y|\mathbf{Z}) \leq Inf_p(X;Y|\mathbf{Z}')$$

Definition 6 (Compound Monotone DAG Faithfulness) *A distribution p is* compound monotone DAG faithful *with respect to a DAG $\mathcal{G}$ if*

$$Active^c_{\mathcal{G}}(X;Y|\mathbf{Z}) \subseteq Active^c_{\mathcal{G}}(X;Y|\mathbf{Z}') \Rightarrow Inf_p(X;Y|\mathbf{Z}) \leq Inf_p(X;Y|\mathbf{Z}')$$

The property is called “monotone” because it states that information in p is a monotonic function of active paths in $\mathcal{G}$. More specifically, monotone DAG faithfulness states that if we do not remove (or “block”) any active paths between two variables in $\mathcal{G}$ by changing the conditioning set, then the information does not decrease. We see that, depending on the definition of an active path, the property can have different consequences. We use $\mathbf{MDF}^s(\mathcal{G})$ and $\mathbf{MDF}^c(\mathcal{G})$ to denote the set of distributions that are monotone DAG faithful with respect to $\mathcal{G}$ using simple and compound active paths, respectively. When we want to avoid specifying the definition of active path, we use $\mathbf{MDF}(\mathcal{G})$ instead.

CGKBL define "monotone DAG faithfulness" only for DAG-perfect distributions, which makes it unclear whether non-DAG-perfect distributions can satisfy this property. In contrast, we define $\mathbf{MDF}^s(\mathcal{G})$ and $\mathbf{MDF}^c(\mathcal{G})$ without reference to other properties of distributions (e.g., DAG-perfect or faithful) in order to analyze the relationship between faithfulness and monotone DAG faithfulness (simple or compound). As previously described, CGKBL use the compound definition of active paths, and thus their definition of monotone DAG faithfulness is precisely our definition of compound monotone DAG faithfulness restricted to distributions that are faithful.

3 Independence-Based and Information-Based Learning Algorithms

In this section, we discuss independence-based and information-based algorithms for learning Bayesian-network structures and discuss the corresponding worst-case running times. Instead of providing formal complexity analyses, which would require us to provide a detailed description of specific instances of these algorithms, we present simple arguments that hopefully will provide the reader with an intuitive understanding of how each type of algorithm handles the most difficult learning scenarios.

In practice, these learning algorithms take an observed set of data and perform statistical tests to evaluate independence and/or mutual information. Thus we can expect the running times of these algorithms to grow with the number of samples in the data. For simplicity, our analysis avoids statistical-sampling issues by effectively assuming that the algorithms have infinite data; each algorithm will have access to an "oracle" that can evaluate independence and/or information as if it had access to the generative distribution. The complexity for an algorithm is then evaluated by the number of times the oracle is called.

3.1 Independence-Based Learning Algorithms

Structure-learning algorithms typically assume that training data is a set of independent and identically distributed samples from some generative distribution p^* that is perfect with respect to some DAG $\mathcal{G}^*$. The goal of the learning algorithm is then to identify $\mathcal{G}^*$ or any DAG that is equivalent to $\mathcal{G}^*$.

A large class of structure-learning algorithms, which we call *independence-based algorithms*, use independence tests to identify and direct edges. If p^* is DAG-perfect and an *independence oracle*—that is, an oracle that provides yes/no answers to queries about conditional independencies in p^*—is available, these algorithms can provably identify a DAG that is equivalent to $\mathcal{G}$ (see, for example, Spirtes, Glymour and Scheines 2000 or Verma and Pearl 1991).

Although many different algorithms have been developed, the basic idea behind independence-based algorithms is as follows. In a first phase, the algorithms identify pairs of variables that must be adjacent in the generative structure. Under the assumption that p^* is DAG-perfect, variables that are adjacent in the generative structure have the property that they are not independent given any conditioning set. The independence oracle is used to check whether this property holds for each pair of variables. Various algorithms provide improvements over an exhaustive search over all subsets of variables. In the second phase, the identified edges are directed.

3.2 Why Independence-Based Learning is Hard

A worst-case scenario for the independence-based algorithms is when the generative structure is as shown in Figure 2 in which all variables are adjacent *except* for A and B. More specifically, (1) the variables in $\mathbf{X} = \{X_1, \ldots, X_n\}$ are parents of A, B and all variables in $\mathbf{Y} = \{Y_1, \ldots, Y_m\}$, (2) both A and B are parents of all the variables in $\mathbf{Y}$, (3) X_i is a parent of X_j for all $i < j$, and (4) Y_i is a parent of Y_j for all $i < j$. For this structure the independence oracle will return "not independent" for *any* test other than "is A independent of B given $\mathbf{X}$?". This extreme example demonstrates that—when using an independence oracle—the only way to determine whether A and B are adjacent is to enumerate and test all possible conditioning sets; using an adversarial argument, we could have the oracle return "not independent" on all but the *last* conditioning set. Because there are $2^{|\mathbf{X}|+|\mathbf{Y}|}$ possible conditioning sets, identifying whether or not the generative network contains an edge between A and B is intractable.

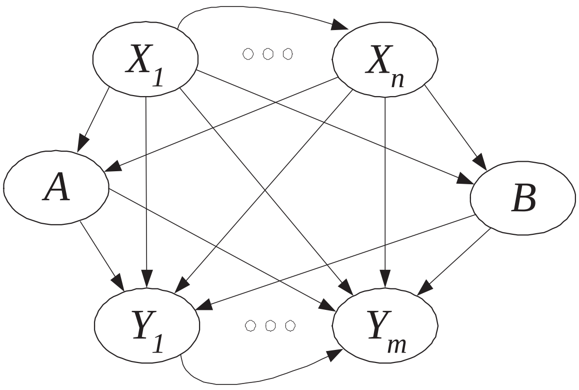

Figure 2: Worst-case scenario for independence-based learning algorithms.

3.3 Information-Based Learning Algorithms

CGKBL take a slightly different approach to learning Bayesian networks. Instead of using conditional independence directly, they use conditional-mutual information both to test for independence and to help guide the learning algorithm. Information can be used to measure the degree of conditional dependence among sets of variables; the following well-known fact about information (e.g., Cover and Thomas, 1991) helps provide insight into this relationship.

Fact 1 $Inf_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) = 0$ *if and only if* $Ind_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$.

Fact 1 demonstrates that any algorithm that utilizes an independence oracle can be modified to use an information oracle. The potential for improvement lies in the fact that we receive additional information when using an information oracle. With this additional information and the MDF assumption, CGKBL claim that their algorithm identifies the generative structure using a polynomial number of queries to the information oracle in the worst case. It turns out that the worst-case scenario considered in the previous section is also the key scenario for information-based algorithms. In particular, it is reasonably easy to show that if we can identify the set $\mathbf{X}$ in Figure 2 efficiently, then we can identify the entire generative structure efficiently.

Consider again the graph in Figure 2. We can use MDF to identify the set $\mathbf{X}$ using the following *greedy* algorithm: start with the conditioning set $\mathbf{S} = \mathbf{X} \cup \mathbf{Y}$, and then repeatedly remove from $\mathbf{S}$ the variable that results in the largest *decrease* in information between A and B, until no removal decreases the information. If the resulting information is zero, we know that there is no edge between A and B; otherwise, we conclude that there is an edge.

A non-rigorous argument for why the greedy algorithm is correct for the example is as follows. First, the algorithm never removes any element of $\mathbf{X}$ from $\mathbf{S}$ because the removal of any such element—when the remaining elements of $\mathbf{X}$ are in $\mathbf{S}$—cannot "block" any active paths (under either definition of an active path) between A and B. Thus, because the number of active paths has necessarily increased, we conclude by MDF that the information in p^* cannot decrease from such a removal. Second, it is possible to show that the deepest variable $Y_i \in \mathbf{Y} \cap \mathbf{S}$ (the variable with the largest index) has the property that if it is removed from $\mathbf{S}$, no active paths are created; thus, because removing Y_i from $\mathbf{S}$ will "block" the previously active path $A \rightarrow Y_i \leftarrow B$, we conclude from MDF that the information cannot increase in p^* from the removal. For simplicity, we ignore the boundary cases where removing a member from either $\mathbf{X}$ or $\mathbf{Y}$ does not change the information; under this scenario (1) the information increases as a result of removing any variable from $\mathbf{X}$, and (2) there is always a variable Y_i from $\mathbf{Y} \cap \mathbf{S}$ such that the information decreases by removing Y_i from $\mathbf{S}$. We conclude that the greedy algorithm will terminate with the correct conditioning set $\mathbf{S} = \mathbf{X}$. Furthermore, each iteration of the algorithm requires at most $|\mathbf{S}| = |\mathbf{X}| + |\mathbf{Y}|$ calls to the information oracle, and there will be $|\mathbf{Y}|$ such iterations. Thus, the greedy algorithm will terminate after $\mathrm{O}(|\mathbf{Y}|^2 + |\mathbf{X}| \cdot |\mathbf{Y}|)$ calls to the information oracle.

CGKBL define a specific information-based learning algorithm that overcomes the worst-case exponential behavior described in the previous section by using a greedy search as above to determine whether or not an edge should be present. Furthermore, they provide a similar argument as above to claim that given $p^* \in \mathbf{Perfect}(\mathcal{G}^*) \cap \mathbf{MDF}(\mathcal{G}^*)$, the algorithm will recover the generative structure (up to equivalence).

If MDF were a reasonable assumption, the result of CGKBL would be significant. Because a mutual-information oracle can be approximated with increasing accuracy as (1) the number training cases increases and (2) the number of variables in the query decreases, we expect that in many real domains we might be able to learn the generative structure—using finite data—in a reasonable amount of time. As we demonstrate in the next section, however, MDF is generally not a reasonable assumption.

4 The Monotone DAG Faithfulness Assumption is not Reasonable

Without studying the details of MDF, the assumption may seem intuitively appealing at first: suppose that removing a variable from the conditioning set "deactivates" some paths between A and B in the generative structure without simultaneously "activating" any other paths. Then we might be tempted to believe that the mutual information between A and B should decrease, or at least not increase. CGKBL state:

> In real world situations most faithful models are also monotone DAG-faithful. We conjecture that the violations of monotone DAG-faithfulness only happen when the probability distributions are 'near' the violations of DAG-faithfulness.

If the CGKBL conjecture were true, it would have significant consequences for learning. First, most structure-learning algorithms assume faithfulness to prove correctness and thus, by assuming a little bit more, we could obtain a polynomial-time algorithm. Second, for a given structure $\mathcal{G}$, almost all distributions in $\mathbf{Markov}(\mathcal{G})$ are faithful, and thus we could be confident that our assumptions are not too limiting.

In this section, we show that in real-world situations, faithfulness and MDF are incompatible with each other, and thus MDF is not a good assumption. Before proving our main result, we find it useful to explore some examples that demonstrate some specific problems with MDF. In Section 4.1, we provide a simple example of a distribution that violates MDF and is not simultaneously "close" to being non-faithful. In Section 4.2, we show a simple example where MDF leads to counterintuitive consequences. In Section 4.3, we prove our main result: unless the generative structure comes from a severely restricted class of models, MDF and faithfulness are incompatible.

4.1 A Simple Violation of MDF

In this section, we demonstrate that the distribution for a Bayesian network need not satisfy the MDF assumption. Consider the Bayesian-network structure shown in Figure 3 and the corresponding set of parameters shown in Table 1. Note that the structure of this example is a particular instance of the worst-case-scenario model from Section 3.2.

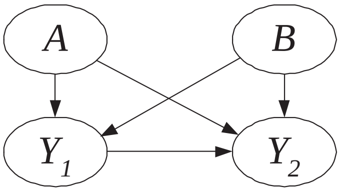

Figure 3: Structure of a Bayesian network that violates the MDF assumption.

Under either definition of MDF, this Bayesian network provides an example of a violation of MDF. In particular, under either definition of an active path, the set of active paths between A and B given both Y_1 and Y_2 is a superset of the set of active paths when only Y_1 is in the conditioning set. Thus, for any distribution p contained in either $\mathbf{MDF}^s(\mathcal{G})$ or $\mathbf{MDF}^c(\mathcal{G})$ we have

$$Inf_p(A; B|Y_1 \cup Y_2) \geq Inf_p(A; B|Y_1).$$

For the joint distribution q obtained from the conditional distributions in the table, however, we have $Inf_q(A; B|Y_1 \cup Y_2) = 0.33$ and $Inf_q(A; B|Y_1) = 0.35$. If we consider

A	$p(A)$
0	0.5
1	0.5

B	$p(B)$
0	0.5
1	0.5

A	B	Y_1	$p(Y_1\|A,B)$
0	0	0	0.38
0	0	1	0.62
0	1	0	0.01
0	1	1	0.99
1	0	0	0.20
1	0	1	0.80
1	1	0	0.99
1	1	1	0.01

A	B	Y_1	Y_2	$p(Y_2\|A,B,Y_1)$
0	0	0	0	0.96
0	0	0	1	0.04
0	0	1	0	0.22
0	0	1	1	0.78
0	1	0	0	0.35
0	1	0	1	0.65
0	1	1	0	0.91
0	1	1	1	0.09
1	0	0	0	0.89
1	0	0	1	0.11
1	0	1	0	0.99
1	0	1	1	0.01
1	1	0	0	0.05
1	1	0	1	0.95
1	1	1	0	0.50
1	1	1	1	0.50

Table 1: Parameters of a Bayesian network that violates the MDF assumption.

the equivalent structure in which the edge between Y_1 and Y_2 is reversed, we obtain the inequality

$$Inf_p(A;B|Y_1 \cup Y_2) \geq Inf_p(A;B|Y_2).$$

Using the same distribution q (which is Markov with respect to the modified structure) we have $Inf_q(A;B|Y_1 \cup Y_2) = 0.33$ and $Inf_q(A;B|Y_2) = 0.40$. Thus in both cases, the distribution q is not contained in either $\mathbf{MDF}^s(\mathcal{G})$ or $\mathbf{MDF}^c(\mathcal{G})$.

Our example is particularly interesting because it illustrates that violations can occur during crucial phases of the CGKBL learning algorithm. Namely, in order for the algorithm to learn that there is no edge between A to B, it must successfully identify the marginal independence. To get to the point where this independence test is made, the algorithm must first find that either $Inf_p(A;B|Y_1 \cup Y_2) > Inf_p(A;B|Y_1)$ or $Inf_p(A;B|Y_1 \cup Y_2) > Inf_p(A;B|Y_2)$, neither of which is true in this example. This failure would lead the algorithm to learn incorrectly that there is an edge between A and B.

CGKBL provide their own counterexample to MDF by first specifying an unfaithful distribution, and then considering slight modifications to the parameters so that the distribution is no longer unfaithful, but it is "close" to being unfaithful. Our example, on the other hand, is not "close" to being unfaithful in the following sense: the important information values above (i.e., 0.33, 0.35 and 0.40) indicate significant dependence according to the threshold used by CGKBL. In particular, CGKBL deem two variables conditionally independent only if the corresponding mutual information is less than either 0.01 or 0.0025 (depending on the experiment).[2] In addition, the pair-wise infor-

[2] CGKBL do not define the base of the logarithm that they use, but two is standard when calculating

A	$p(A)$
0	0.5
1	0.5

B	$p(B)$
0	0.6
1	0.4

A	B	C	$p(C\|AB)$
0	0	0	0.99
0	0	1	0.01
0	1	0	0.5
0	1	1	0.5
1	0	0	0.99
1	0	1	0.01
1	1	0	0.99
1	1	1	0.01

C	D	$p(D\|C)$
0	0	0.9
0	1	0.1
1	0	0.01
1	1	0.99

Table 2: Parameters of a Bayesian network—whose structure is given in Figure 1—for which MDF leads to counterintuitive conclusions.

mation values corresponding to all of the edges in the structure also indicate significant dependence.

4.2 Counterintuitive Consequences of MDF

Consider the DAG model shown in Figure 1. The two definitions of MDF (simple and compound) correspond to two different sets of distributions for this example. In particular, for the simple definition, we have $Active^s_{\mathcal{G}}(A;B|C) = Active^s_{\mathcal{G}}(A;B|D)$ and thus $Inf_p(A;B|C) = Inf_p(A;B|D)$ for any distribution p in $\mathbf{MDF}^s(\mathcal{G})$. For the compound definition, we have $Active^c_{\mathcal{G}}(A;B|C) \subseteq Active^c_{\mathcal{G}}(A;B|D)$ and thus $Inf_p(A;B|C) \leq Inf_p(A;B|D)$ for any distribution p in $\mathbf{MDF}^c(\mathcal{G})$.

The equality of the information for distributions in $\mathbf{MDF}^s(\mathcal{G})$ is *a priori* unreasonable. The inequality for distributions in $\mathbf{MDF}^c(\mathcal{G})$, on the other hand, seems to be counterintuitive. That is, it seems plausible that there should be *more* dependence between A and B when given C than when given D, and thus we might expect an information inequality in the opposite direction than what holds in $\mathbf{MDF}^c(\mathcal{G})$. Rather surprising, this inequality can be satisfied using the conditional distributions in Table 2. For this distribution, the difference $Inf_p(A;B|C) - Inf_p(A;B|D) = -0.006$.

To help understand how often the information inequality implied by the compound version of MDF occurs, we performed a simple simulation study in which we randomly sampled distributions that are Markov with respect to the structure in Figure 1—where each variable was binary—and computed $Inf_p(A;B|C) - Inf_p(A;B|D)$ for each sampled distribution p. We defined "zero" to be (a conservative) 0 ± 10^{-8} to make sure we did not miss any equalities due to numerical imprecision. Our experiment using 100,000 sampled distributions yielded the following results for the information differences: (a) positive in $99,969$ samples, (b) negative in 31 samples, and (c) "zero" in 0 samples. For this experiment, most of the sampled distributions violate both versions of the MDF assumption. We were surprised by both the existence and the frequency of sampled distributions in which the difference $Inf_p(A;B|C) - Inf_p(A;B|D)$ was negative.

information. The other natural candidate bases are e and 10, both of which lead to significant differences according to the CGKBL thresholds.

4.3 Incompatibility of MDF and Faithfulness

In this section, we prove that MDF and faithfulness are incompatible. Before proceeding, we present the following "axiom" that follows from MDF for DAG-perfect distributions, the proof of which is given in the appendix.

Theorem 1 *Let $\mathcal{G}$ be any DAG and let p be any distribution in $\mathbf{MDF}(\mathcal{G}) \cap \mathbf{Perfect}(\mathcal{G})$, where $\mathbf{MDF}(\mathcal{G})$ is defined using either of the two definitions of an active path.*

$$Dsep_{\mathcal{G}}(X;Y|\mathbf{V}) \Rightarrow Ind_p(X;Y)$$

In other words, if two variables are d-separated given *any* conditioning set in $\mathcal{G}$, then for all distributions in $\mathbf{MDF}(\mathcal{G}) \cap \mathbf{Perfect}(\mathcal{G})$, those variables are *marginally* independent. To understand the implication of this result, we define what it means for a DAG *to have a chain*:

Definition 7 (DAG $\mathcal{G}$ has a chain) *A DAG $\mathcal{G}$* has a chain *if one of the following three sub-graphs occurs in $\mathcal{G}$*

- $X \rightarrow Z \rightarrow Y$
- $X \leftarrow Z \rightarrow Y$
- $X \leftarrow Z \leftarrow Y$.

In other words, a graph has a chain if there is a length-two path between non-adjacent variables that is not a "v-structure" $X \rightarrow Z \leftarrow Y$. The following result, proved by Verma and Pearl (1991), will be useful for proving our main result.

Lemma 1 (Verma and Pearl, 1991) *Let X and Y be non-adjacent variables in DAG $\mathcal{G}$, and let $\mathbf{Z}$ denote the union of their parents. Then $Dsep_{\mathcal{G}}(X;Y|\mathbf{Z})$.*

For the convenience of the interested reader, we provide a proof of Lemma 1 in the appendix. We now prove the main result of this paper:

Theorem 2 *The following statements are jointly inconsistent:*

- *$\mathcal{G}$ has a chain*
- $p \in \mathbf{Perfect}(\mathcal{G})$
- $p \in \mathbf{MDF}(\mathcal{G})$.

where $\mathbf{MDF}(\mathcal{G})$ is defined using either of the two definitions of an active path.

Proof: Suppose $\mathcal{G}$ has a chain, and let p be any distribution in $\mathbf{MDF}(\mathcal{G}) \cap \mathbf{Perfect}(\mathcal{G})$. By definition of a chain, there exists a non-adjacent pair of variables X and Y in $\mathcal{G}$ that are connected by a length-two path through Z, where Z is a parent of either X or Y (or both). From Lemma 1, we know $Dsep_{\mathcal{G}}(X;Y|\mathbf{Z})$ where $\mathbf{Z}$ is the union of the parents of X and Y in $\mathcal{G}$. From Theorem 1, this implies that $Ind_p(X;Y)$. But the length-two path between X and Y through Z constitutes a simple (and compound) active path in $\mathcal{G}$ given the empty conditioning set, and we conclude that p is not faithful to $\mathcal{G}$, contradicting the supposition that $p \in \mathbf{Perfect}(\mathcal{G})$. ■

One possible "fix" in light of this negative result would be to weaken the requirement that p be faithful. As described in Section 2.2, however, almost all distributions in $\mathbf{Markov}(\mathcal{G})$ are also in $\mathbf{Faithful}(\mathcal{G})$, so we can conclude that for generative structures that have chains, the MDF assumption is not reasonable. The only hope for the MDF assumption is that it proves to be useful in some (unrealistic) learning scenario where we can assume that the generative distribution is perfect with respect to some DAG with no chain. As we saw in Section 4.1, the assumption can be violated for such a distribution, but given the $\mathrm{O}(n^4)$ result of CGKBL it might be worth restricting the possible generative distributions. In this scenario, however, it is easy to derive an independence-based learning algorithm that (1) does not need to assume MDF and (2) identifies the optimal structure in just $\mathrm{O}(n^2)$ calls to an independence oracle! In particular, because we know ahead of time that only marginal independence facts hold in the generative distribution, we can identify all of them by testing, for each pair of variables A and B, whether $Ind_p(A; B)$. After all the independence facts have been identified, we direct all the edges using standard approaches. Thus there seems to be no benefit from assuming MDF.

5 Discussion

In this paper, we demonstrated that the monotone DAG faithfulness assumption is incompatible with the faithfulness assumption unless we are in an unrealistic learning scenario where the optimal structure can be identified with $\mathrm{O}(n^2)$ calls to an independence oracle. Unfortunately, this means that the optimality guarantees of the CGKBL algorithm are valid only in unrealistic situations where a faster learning algorithm is also optimal. Furthermore, because an independence oracle can be implemented with an information oracle, the faster algorithm requires a less powerful oracle.

Given the unreasonable consequences of MDF, it is intriguing that the assumption is so intuitively appealing. We believe that the source of the misguided intuition stems from the fact that—assuming faithfulness—information is zero if and only if there are no active paths. In particular, this fact implies that for any faithful distribution, the "information flow" between two variables necessarily increases when the set of active paths changes from the empty set to something other than the empty set. The mistake is to extrapolate from this base case and conclude that a non-zero "information flow" does not decrease when we add (zero or more elements) to the set of active paths.

Our study of MDF has led to a surprising result about the structure in Figure 1. In distributions faithful to that structure, the conditional mutual information between A and B can be *larger* when given D than when given C. Although we found that such distributions were not common given our sampling scheme, they occurred regularly enough that they cannot be discarded as anomalous.

Although MDF is a bad assumption, CGKBL have brought up an interesting question: can we make some connection between active paths and information that might lead to more efficient learning algorithms? Perhaps a modified definition of an active path—that is equivalent to the ones discussed here in terms of the d-separation criterion—would yield more realistic constraints on distributions and yet still lead to an efficient algorithm.

References

[Cheng et al., 2002] Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137:43–90.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* John Wiley and Sons, Inc., New York.

[Meek, 1995] Meek, C. (1995). Strong-completeness and faithfulness in belief networks. In Hanks, S. and Besnard, P., editors, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence,* Montreal, QU, pages 411–418. Morgan Kaufmann.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

[Spirtes et al., 2000] Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search (second edition).* The MIT Press, Cambridge, Massachussets.

[Verma and Pearl, 1991] Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. In Henrion, M., Shachter, R., Kanal, L., and Lemmer, J., editors, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227.

Appendix: Proofs

In this appendix, we prove Proposition 1, Lemma 1, and Theorem 1. We begin by proving three propositions about active paths, the first of which is Proposition 1.

Proposition 1 *There is a simple active path between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$ if and only if there is a compound active path between X and Y given conditioning set $\mathbf{Z}$ in $\mathcal{G}$.*

Proof: Because a simple active path is also a compound active path, we need only show the existence of a simple active path between X and Y given a compound active path π between X and Y. We establish this result by showing that any sub-path of π that begins and ends with the same variable W may be "skipped" by replacing the entire sub-path with the single variable W, such that the resulting path π' remains active; after repeatedly removing all such sub-paths, the resulting (active) path will necessarily be simple. It is easy to see that after removing the sub-path from W to itself from π, the two properties of an active path in Definition 1 continue to hold for all variable/positions *other* than the variable W at the position where the sub-path was removed.

To complete the proof, we consider the following three cases:

(a) If $W \in \mathbf{Z}$, then W must be a collider at every position along π, and therefore in π' both edges incident to W at the position where the sub-path was removed are directed into W; thus, because W is a collider at this position in π, it satisfies condition (2) of an active path in Definition 1.

(b) If $W \notin \mathbf{Z}$, and W is not a collider in π' at the position where the sub-path was removed, then W at this point satisfies condition (1) of Definition 1.

(c) The final case to consider is if $W \notin \mathbf{Z}$ and W is a collider in π' at the position the sub-path was removed. Consider a traversal of the sub-path from W to itself that was removed from π to produce π'. If either the first or the last edge in this path is directed *toward* W, then W is a collider at that point in π, from which we conclude that W has a descendant in $\mathbf{Z}$ and thus W satisfies condition (2) of Definition 1. Otherwise, the first and last edge of this W to W path are both directed *away* from W. This means that at some point along the traversal, we must hit some collider on π that satisfies condition (2) of Definition 1. Because the *first* such collider is a descendant of W, condition (2) of Definition 1 is satisfied for W at this position in π', and thus the proposition follows. ■

Proposition 2 *For either definition of an active path, if $\pi \in Active_{\mathcal{G}}(Z;Y|\mathbf{W})$ but $\pi \notin Active_{\mathcal{G}}(Z;Y|\mathbf{W} \cup X)$ then X occurs as a non-collider at some position in π.*

Proof: We know that π is active when the conditioning set is $\mathbf{W}$, but if we add X to the conditioning set, π is no longer active. Therefore, from Definition 1, after adding X to the conditioning set either (1) there is a non-collider on the path that is now in the conditioning set, or (2) there is a collider on the path that—after the addition—is not in the conditioning set and has no descendants in the conditioning set. Because no variables were removed from the conditioning set, we know that only (1) is possible and that X is a non-collider at some position on π. ■

Proposition 3 *Let $\pi = \{A_{(1)}, \ldots, A_{(n)}\}$ be any path in $Active^c_{\mathcal{G}}(Z;Y|\mathbf{W})$. Then for any $A_{(i)}$ and $A_{(j)}$ such that $i < j$, $A_{(i)} \notin \mathbf{W}$ and $A_{(j)} \notin \mathbf{W}$, the sub-path $\pi' = \{A_{(i)}, \ldots, A_{(j)}\}$ is in $Active^c_{\mathcal{G}}(A_{(i)}; A_{(j)}|\mathbf{W})$.*

Proof: From Definition 1, all variables in π' satisfy one of the two necessary conditions, with the possible exception of the endpoints; these variables can be colliders in the original path, but are necessarily non-colliders (see the definition of a *collider at a position* in Section 2.1) in π'. Because neither endpoint is in $\mathbf{W}$, condition (1) in Definition 1 is satisfied for the endpoints and the proposition follows. ■

Proposition 3 simply asserts that any sub-path of an active path between two variables that are not in the conditioning set is itself active. Because a simple path is a compound path, the proposition also holds for simple active paths.

Lemma 1 (Verma and Pearl, 1991) *Let X and Y be non-adjacent variables in DAG $\mathcal{G}$, and let $\mathbf{Z}$ denote the union of their parents. Then $Dsep_{\mathcal{G}}(X;Y|\mathbf{Z})$.*

Proof: Suppose that X and Y are not adjacent in DAG $\mathcal{G}$ but that there is a simple active path π between X and Y given $\mathbf{Z}$. Because $\mathbf{Z}$ contains the parents of both X and Y, the variable immediately following X (proceeding Y) must be a descendant of X (Y). It follows that there must be a collider at some position on path π. Furthermore, the collider at the position nearest to X on π is a descendant of X and, similarly, the collider at the position nearest to Y on π is a descendant or Y. For the path π to be active, however, these colliders must be in $\mathbf{Z}$ or have descendants in $\mathbf{Z}$, which would imply the existence of a directed cycle and thus a contradiction. ∎

Note that the next lemma is relevant to *simple* active paths.

Lemma 2

$$Dsep_{\mathcal{G}}(X; Y|\mathbf{W} \cup Z) \Rightarrow Active^{s}_{\mathcal{G}}(Z; Y|\mathbf{W}) \subseteq Active^{s}_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$$

Proof: If either X or Y is an element of $\mathbf{W}$, the lemma follows easily; for the remainder of the proof we assume that neither variable is contained in the conditioning set. Suppose $Dsep_{\mathcal{G}}(X; Y|\mathbf{W} \cup Z)$ and there exists a path π in $Active^{s}_{\mathcal{G}}(Z; Y|\mathbf{W})$ that is not in $Active^{s}_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$. From Proposition 2, we conclude that X must be a non-collider at some position i along π. We now consider the sub-path π' of π that starts at variable X in position i, and continues to variable Y. Because neither X nor Y is in $\mathbf{W}$, we know from Proposition 3 that $\pi' \in Active_{\mathcal{G}}(X; Y|\mathbf{W})$. Furthermore, because π is a *simple* path that starts at Z, we know that π' does not contain Z and consequently must be contained in $Active_{\mathcal{G}}(X; Y|\mathbf{W} \cup Z)$. But this contradicts the supposition $Dsep_{\mathcal{G}}(X; Y|\mathbf{W} \cup Z)$. ∎

We find it convenient to use $\mathbf{Pa}^{\mathcal{G}}_{X}$ to denote the set of parents of variable X in DAG $\mathcal{G}$.

Lemma 3 *Let X and Y be any pair of variables that are not adjacent in $\mathcal{G}$, and for which X is not an ancestor of Y, and let $\mathbf{D}$ be any non-empty subset of $\{\mathbf{Pa}^{\mathcal{G}}_{X} \cup \mathbf{Pa}^{\mathcal{G}}_{Y}\}$ such that $Dsep_{\mathcal{G}}(X; Y|\mathbf{D})$. Let $\mathbf{W} = \mathbf{D} \setminus Z$, for any variable $Z \in \mathbf{D}$. Then under either of the two definitions of an active path*

$$Active_{\mathcal{G}}(Z; Y|\mathbf{W}) \subseteq Active_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$$

Proof: Because $Dsep_{\mathcal{G}}(X; Y|\mathbf{W} \cup Z)$, the lemma follows immediately from Lemma 2 for the simple definition of an active path. For the remainder of the proof, we consider only the compound definition of an active path.

We prove the lemma by contradiction. In particular, we show that if there exists an active path in $Active^{c}_{\mathcal{G}}(Z; Y|\mathbf{W})$ that is not in $Active^{c}_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$, then there exists some $W \in \mathbf{W}$ that is a descendant of X in $\mathcal{G}$. Identifying such a W yields a contradiction by the following argument: if W is a parent of X, then we have identified a directed cycle in $\mathcal{G}$, and if W is a parent of Y, then X is an ancestor of Y.

The remainder of the proof demonstrates the existence of $W \in \mathbf{W}$ that is a descendant of X in $\mathcal{G}$. Let $\pi = \{A_{(1)}, \ldots, A_{(n)}\}$—where $A_{(1)} = Z$ and $A_{(n)} = Y$—be any path in $Active^{c}_{\mathcal{G}}(Z; Y|\mathbf{W})$ such that $\pi \notin Active^{c}_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$. From Proposition 2, X must appear as a non-collider at some position i along π; that is, $A_i = X$ and π must contain one of the following three sub-paths:

1. $A_{(i-1)} \rightarrow X \rightarrow A_{(i+1)}$
2. $A_{(i-1)} \leftarrow X \leftarrow A_{(i+1)}$
3. $A_{(i-1)} \leftarrow X \rightarrow A_{(i+1)}$

We now consider any path π' that starts at X and then follows the edges in π (toward either $A_{(1)}$ or $A_{(n)}$) such that the first edge is directed *away* from X. That is, if π contains sub-path (1) above we have

$$\pi' = X \rightarrow A_{(i+1)} - \ldots - A_{(n)}$$

where '$-$' denotes an edge in the path without specifying its direction. Similarly, if π contains sub-path (2) above we have

$$\pi' = X \rightarrow A_{(i-1)} - \ldots - A_{(1)}$$

Finally, if π contains sub-path (3) above, π' can be defined as either of the previous two paths.

To simplify our arguments, we rename the elements of π' as follows:

$$\pi' = X \rightarrow B_{(1)} - \ldots - B_{(m)}$$

Consider a traversal of π', starting at the first element X and continuing through each element $B_{(i)}$ for increasing i. If the traversal ever encounters variable Y, it must *first* encounter variable Z; if not, the sub-path from X to Y would constitute an active path that remains active when Z is in the conditioning set, which contradicts the fact that $Dsep_{\mathcal{G}}(X; Y|\mathbf{D})$. Because the last element of the path ($B_{(m)}$) is by definition either Z or Y, we conclude there exists a sub-path π'' of π'

$$\pi'' = X \rightarrow B_{(1)} - \ldots - B_{(r)} - Z$$

that does not pass through variable Y. We know that there must be some edge in π'' that is directed as $B_{(j)} \leftarrow B_{(j+1)}$. Otherwise, there would be a directed path from X to Z in $\mathcal{G}$; if Z is a parent of X this would mean $\mathcal{G}$ contains a cycle, and if Z is a parent of Y this would mean X is an ancestor of Y. Without loss of generality, let $B_{(j)} \leftarrow B_{(j+1)}$ be the *first* edge so directed:

$$\pi'' = X \rightarrow B_{(1)} \rightarrow \ldots \rightarrow B_{(j-1)} \rightarrow B_{(j)} \leftarrow B_{(j+1)} - B_{(j+2)} - \ldots - B_{(r)} - Z$$

Because π'' is a sub-path of π—and because neither endpoint X nor endpoint Z is an element in $\mathbf{W}$—we know from Proposition 3 that it is active given conditioning set $\mathbf{W}$, and thus because it contains the collider $B_{(j-1)} \rightarrow B_{(j)} \leftarrow B_{(j+1)}$, we know from Definition 1 that there is a $W \in \mathbf{W}$ such that either $B_{(j)} = W$ or $B_{(j)}$ is an ancestor of W. Because $B_{(j)}$ is a descendant of X, it follows that W is also a descendant of X, and the proof is complete. ■

The following three facts about mutual information are well-known. See, for example, Cover and Thomas (1991).

Fact 1 *$Inf_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) = 0$ if and only if $Ind_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$.*

Fact 2 *For any p, $Inf_p(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) \geq 0$ for all $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$.*

The last fact is known as the *chain rule* for mutual information.

Fact 3

$$Inf_p(Y; X_1 \cup \ldots \cup X_n|\mathbf{W}) = \sum_{i=1}^{n} Inf_p(Y; X_i|X_1 \cup \ldots \cup X_{i-1} \cup \mathbf{W}).$$

Lemma 4 *Let p be any distribution. Then*

$$Ind_p(X; Y|\mathbf{W} \cup Z) \Rightarrow Inf_p(Z; Y|\mathbf{W}) - Inf_p(Z; Y|\mathbf{W} \cup X) = Inf_p(X; Y|\mathbf{W})$$

Proof: We expand the quantity $Inf_p(Y; X \cup Z|\mathbf{W})$ using the chain rule with two different orders for the variables to obtain

$$Inf_p(Z; Y|\mathbf{W}) + Inf_p(X; Y|\mathbf{W} \cup Z) = Inf_p(X; Y|\mathbf{W}) + Inf_p(Z; Y|\mathbf{W} \cup X)$$

From the independence assumption and Fact 1 we have $Inf_p(X; Y|\mathbf{W} \cup Z) = 0$ and the lemma is established. ■

Finally, we can prove the theorem.

Theorem 1 *Let $\mathcal{G}$ be any DAG and let p be any distribution in* $\mathbf{MDF}(\mathcal{G}) \cap \mathbf{Perfect}(\mathcal{G})$*, where* $\mathbf{MDF}(\mathcal{G})$ *is defined using either of the two definitions of an active path.*

$$Dsep_{\mathcal{G}}(X; Y|\mathbf{V}) \Rightarrow Ind_p(X; Y)$$

Proof: Suppose this is not the case, and that $Dsep_{\mathcal{G}}(X; Y|\mathbf{V})$ but there exists some $p \in \mathbf{MDF}(\mathcal{G}) \cap \mathbf{Perfect}(\mathcal{G})$ in which X and Y are not marginally independent. Because X and Y are d-separated given $\mathbf{V}$, we know that X and Y are not adjacent in $\mathcal{G}$ and thus by Lemma 1 they are d-separated given $\mathbf{Pa}_X^{\mathcal{G}} \cup \mathbf{Pa}_Y^{\mathcal{G}}$. Let $\mathbf{D}$ be any minimal subset of $\mathbf{Pa}_X^{\mathcal{G}} \cup \mathbf{Pa}_Y^{\mathcal{G}}$ for which $Dsep_{\mathcal{G}}(X; Y|\mathbf{D})$; by *minimal* we mean no proper subset of $\mathbf{D}$ also satisfies this property. We know that $\mathbf{D}$ has at least one element because otherwise, by virtue of the fact that $p \in \mathbf{Perfect}(\mathcal{G}) \subseteq \mathbf{Markov}(\mathcal{G})$, X and Y would be marginally independent.

Because $\mathcal{G}$ is a DAG, we know that X and Y cannot be ancestors of each other and thus, without loss of generality, we assume that X is not an ancestor of Y. Let Z be any element of $\mathbf{D}$. From Lemma 3, we know that for $\mathbf{W} = \mathbf{D} \setminus Z$, we have

$$Active_{\mathcal{G}}(Z; Y|\mathbf{W}) \subseteq Active_{\mathcal{G}}(Z; Y|\mathbf{W} \cup X)$$

and thus because $p \in \mathbf{MDF}(\mathcal{G})$, it follows that $Inf_p(Z; Y|\mathbf{W}) \leq Inf_p(Z; Y|\mathbf{W} \cup X)$, or equivalently, $Inf_p(Z; Y|\mathbf{W}) - Inf_p(Z; Y|\mathbf{W} \cup X) \leq 0$. From Lemma 4, however, it follows that this difference is equal to $Inf_p(X; Y|\mathbf{W})$; because information is non-negative (Fact 2), it follows that $Inf_p(X; Y|\mathbf{W}) = 0$ and we conclude from Fact 1 that $Ind_p(X; Y|\mathbf{W})$. Because $\mathbf{W}$ is a proper subset of $\mathbf{D}$, we know from the minimality of $\mathbf{D}$ that p cannot be perfect, yielding a contradiction. ■