A Study Of Identifiability In Causal Bayesian Networks[1]
Version 0.3

Yimin Huang
huang6@cse.sc.edu

Marco Valtorta
mgv@cse.sc.edu

Department of Computer Science and Engineering
University of South Carolina

February 20, 2006

[1]University of South Carolina Department of Computer Science and Engineering Report CSE TR-2006-002

Abstract

This paper addresses the problem of identifying causal effects from nonexperimental data in a *causal Bayesian network*, i.e., a directed acyclic graph that represents causal relationships. The identifiability question asks whether it is possible to compute the probability of some set of (effect) variables given intervention on another set of (intervention) variables, in the presence of non-observable (i.e., hidden or latent) variables. It is well known that the answer to the question depends on the structure of the causal Bayesian network, the set of observable variables, the set of effect variables, and the set of intervention variables. Our work is based on the work of Tian and Pearl [1, 2, 3] and our own work [4], and extends it. We show that the *identify* algorithm that Tian and Pearl define and prove sound for semi-Markovian models can be transfered to general causal graphs and is not only sound, but also complete. This result effectively solves the identifiability question for causal Bayesian networks that Pearl posed in 1995 [5], by providing a sound and complete algorithm for identifiability.

1 Introduction

This paper focuses on the feasibility of inferring the strength of cause-and-effect relationships from a causal graph [5] [6], which is an acyclic directed graph expressing nonexperimental data and causal relationships. Because of the existence of unmeasured variables, the following identifiability questions arise: "Can we assess the strength of causal effects from nonexperimental data and casual relationships? And if we can, what is the total causal effect in terms of estimable quantities?"

The questions just given can partially be answered using a graphical approach due to Pearl and his collaborators. More precisely, graphical conditions have been devised to show whether a causal effect, that is, the joint response of any set S of variables to interventions on a set T of action variables, denoted $P_T(S)$ [1] is identifiable or not. Those results are summarized in [6]. For example, "back-door" and "front-door" criteria and do-calculus [5]; graphical criteria to identify $P_T(S)$ when T is a singleton [7]; graphical conditions under which it is possible to identify $P_T(S)$ where T and S are, possibly non-singleton, sets, subject to a special condition called Q-identifiability [8]. Some further study can be also found in [9] and [10].

Recently, J. Tian & J. Pearl and also J. Tian himself published a series of papers related to this topic [1, 2, 3, 11]. Their new methods combined the graphical character of causal graph and the algebraic definition of causal effect. They used both algebraic and graphical methods to identify causal effects. The basic idea is, first, to transfer causal graphs to semi Markovian graphs [2], then to use Algorithm 2 in [3] (henceforth, the *Identify* algorithm) to calculate the causal effects we want to know.

Tian and Pearl's method was a great contribution to this study area. But there were still some problems left. First, even though we believe, as Tian and Pearl do, that the semi Markovian models obtained from the transforming Projection algorithm in [2] are equal to the original causal graphs, and therefore the causal effects should be the same in both models, still, to the best of our knowledge, there is no formal proof for this equivalence. Second, the completeness question of the Identify algorithm in [3] was still open, so that it was unknown whether a causal effect is identifiable if the Identify algorithm 2 fails.

Following Tian and Pearl's work, Huang and Valtorta [4] solved the second question. They showed that the Identify algorithm 2 Tian and Pearl used on semi Markovian models is sound and complete. Shpitser and Pearl recently obtained a similar result independently [12].

In this paper, we focus on general causal graphs directly and our proofs show, as Tian and Pearl pointed out, that Algorithm 2 in [3] can also be used in general causal models. After that, we prove that the algorithm is complete, which means a causal effect is identifiable if and only if the given algorithm runs successfully and returns an expression which is the target causal effect in terms of estimable quantities.

[1] Pearl and Tian used notation $P(s|do(t))$ and $P(s|\hat{t}\,)$ in [6] and $P_t(s)$ in [2], [3].

In the next section we present the definitions and notations used in this paper. In section three, we repeat some important lemmas that will be used to support the identify algorithm. We prove that an algorithm for a special kind of identiafibility question, called $Q[S]$, is sound and complete in section four. Based on this result, in section five, we present a version of the identify algorithm that can work on any causal graph. We also prove that this algorithm is sound and complete. Conclusions are in section six. Proof of the correctness of the lemmas we used in this article is given in the appendixes.

2 Definitions and Notations

Markovian models are popular graphical models for encoding distributional and causal relationships. A *Markovian model* consists of a DAG G over a set of variables $V = \{V_1, \ldots, V_n\}$, called a *causal graph* and a probability distribution over V, which has some constraints on it that will be specified precisely below. We use $V(G)$ to show that V is the variable set of graph G. If it is clear in the context, we also use V directly. The interpretation of such kind of model consists of two parts. The first one says that each variable in the graph is independent of all its non-descendants given its direct parents. The second one says that the directed edges in G represent causal influences between the corresponding variables. A Markovian model for which only the first constraint holds is called a *Bayesian network*. This explains why Markovian models are also called *causal Bayesian networks*. Some authors prefer to consider equation 3 (below) as definitional; others take equation 3 as following from more general considerations about causal links. See [13] and [6].

In this paper, capital characters, like V, are used for variable sets; the lower characters, like v, stand for the instances of variable set V. Capital character like X, Y and V_i are also used for single variable, and their values can be x, y and v_i. Normally, we use $F(V)$ to denote a function on variable set V. An instance of this function is denoted as $F(V)(V = v)$, or $F(V)(v)$, or just $F(v)$. Because all the variables are in the causal graph, we sometimes use node or node set instead of variable and variable set.

As in most work on Bayesian networks and, more generally, on directed graphs, we use $Pa(V_i)$ to denote parent node set of variable V_i in graph G and $pa(V_i)$ to denote an instance of $Pa(V_i)$. $Ch(V_i)$ is V_i's children node set; $ch(V_i)$ is an instance of $Ch(V_i)$.

Based on the probabilistic interpretation, we get that the joint probability function $P(v) = P(v_1, \ldots, v_n)$ can be factorized as

$$P(v) = \prod_{V_i \in V} P(v_i | pa(V_i)) \tag{1}$$

The causal interpretation of Markovian model enables us to predict the intervention effects. Here, intervention means some kind of modification of factors in product (1). The simplest kind of intervention is fixing a subset $T \subseteq V$

of variables to some constants t,denoted by $do(T = t)$ or just $do(t)$, and then the post-intervention distribution

$$P_T(V)(T = t, V = v) = P_t(v) \tag{2}$$

is given by:

$$P_t(v) = P(v|do(t)) = \begin{cases} \prod_{V_i \in V \setminus T} P(v_i|pa(V_i)) & v \text{ consistent with } t \\ 0 & v \text{ inconsistent with } t \end{cases} \tag{3}$$

We note explicitly that the post-intervention distribution $P_T(V)(T = t, V = v) = P_t(v)$ is a probability distribution.

When all the variables in V are observable, since all $P(v_i|pa(V_i))$ can be estimated from nonexperimental data, as just indicated, all causal effects are computable. But when some variables in V are unobservable, things are much more complex.

Let $N(G)$ and $U(G)$ (or simply N and U when the graph is clear from the context) stand for the sets of observable and unobservable variables in graph G respectively, that is $V = N \cup U$. The observed probability distribution $P(n) = P(N = n)$, is a mixture of products:

$$P(n) = \sum_{U_k \in U} \prod_{V_i \in V} P(v_i|pa(V_i)) = \sum_{U_k \in U} \prod_{V_i \in N} P(v_i|pa(V_i)) \prod_{V_j \in U} P(v_j|pa(V_j)) \tag{4}$$

The post-intervention distribution $P_t(n) = P_{T=t}(N = n)$ [2] is defined as:

$$P_t(n) = \begin{cases} \sum_{U_k \in U} \prod_{V_i \in N \setminus T} P(v_i|pa(V_i)) & \prod_{V_j \in U} P(v_j|pa(V_j)) \\ & n \text{ consistent with } t \\ 0 & n \text{ inconsistent with } t \end{cases} \tag{5}$$

Sometimes what we want to know is not the post-intervention distribution for the whole N, but the post-intervention distribution $P_t(s)$ of an observable variable subset $S \subset N$. For those two observable variable set S and T, $P_t(s) = P_{T=t}(S = s)$ is given by:

$$P_t(s) = \begin{cases} \sum_{V_l \in (N \setminus S) \setminus T} \sum_{U_k \in U} \prod_{V_i \in N \setminus T} P(v_i|pa(V_i)) & \prod_{V_j \in U} P(v_j|pa(V_j)) \\ & s \text{ consistent with } t \\ 0 & s \text{ inconsistent with } t \end{cases} \tag{6}$$

The identifiability question is defined as whether the causal effect $P_T(S)$, that is all $P_t(s)$ given by (6), can be determined uniquely from the distribution $P(N = n)$ given by (4), and thus independent of the unknown quantities $P(v_i|pa(V_i))$s, where $V_i \in U$ or there are some $V_j \in Pa(V_i)$, $V_j \in U$.

[2] In this paper, we only consider the situation in which $T \subseteq N$.

We give out a formal definition of *identifiability* below, which follows [3].

A Markovian model consists of four elements

$$M =< N, U, G_{N\cup U}, P(v_i|pa(V_i)) >$$

where, (i) N is a set of observable variables; (ii) U is a set of unobservable variables; (iii) G is a directed acyclic graph with nodes corresponding to the elements of $V = N \cup U$; and (vi) $P(v_i|pa(V_i))$, is the conditional probability of variable $V_i \in V$ given its parents $Pa(V_i)$)in G.

Definition 1 The causal effect of a set of variables T on a disjoint set of variables S is said to be identifiable from a graph G if all the quantities $P_t(s)$ can be computed uniquely from any positive probability of the observed variables — that is , if $P_t^{M_1}(s) = P_t^{M_2}(s)$ for every pair of models M_1 and M_2 with $P^{M_1}(n) = P^{M_2}(n) > 0$ and $G(M_1) = G(M_2)$.

This definition means that, given the causal graph G, the quantity $P_t(s)$ can be determined from the observed distribution $P(n)$ alone; the details of M are irrelevant.

Normally, when we talk about S and T, we think they are both observable variable subsets of N and mutually disjoint. So, equation 6 can be replaced by

$$P_t(s) = \sum_{V_l\in(N\backslash S)\backslash T} \sum_{U_k\in U} \prod_{V_i\in N\backslash T} P(v_i|pa(V_i)) \prod_{V_j\in U} P(v_j|pa(V_j)) \quad (7)$$

From now on, we will use this definition instead of equation 6.

We are sometimes interested in the causal effect on a set of observable variables S due to all other observable variables. In this case, keeping the convention that N stands for the set of all observable variables and T stands for the set of variables whose effect we want to compute, $T = N\backslash S$, and equation 7 simplifies to

$$P_{n\backslash s}(s) = \sum_{u_k\in U} \prod_{V_i\in S} P(v_i|pa(V_i)) \prod_{V_j\in U} P(v_j|pa(V_j)) \quad (8)$$

In formula 8, the subscript $n\backslash s$ indicates a configuration of the variable or variables in the set $N\backslash S$. For convenience and for uniformity with [2], we define

$$Q[S] = P_{N\backslash S}(S) \quad (9)$$

and interpret this equation as stating that $Q[S]$ is the causal effect of $N\backslash S$ on S.

Note that $Q[S]$ is identifiable if $Q[S]^{M_1}(s) = Q[S]^{M_2}(s)$ for every pair of models M_1 and M_2 with $Q[N]^{M_1}(n) = Q[N]^{M_2}(n) > 0$ and $G(M_1) = G(M_2)$.

We define the *c-component relation* on the unobserved variable set U of graph G as:

For any $U_1 \in U$ and $U_2 \in U$, they are related under the c-component relation if and only if at least one of conditions below is satisfied:

(i) there is an edge between U_1 and U_2,

(ii) U_1 and U_2 are both parents of the same observable node,

(iii) both U_1 and U_2 are in the c-component relation with respect to another node $U_3 \in U$.

Observe that the c-component relation in U is reflexive, symmetric and transitive, so it defines a partition of U. Based on this relationship, we can therefore divide U into disjoint and mutually exclusive c-component related parts.

A *c-component* of variable set V on graph G consists of all the unobservable variables belonging to the same c-component related part of U and all observable variables that have an unobservable parent which is a member of that c-component. According to the definition of c-component relation, it is clear that an observable node can only appear in one c-component. If an observable node has no unobservable parent, then it is a c-component on V by itself. Therefore, the c-components form a partition on all of the variables.

For any pair of variables V_1 and V_2 in causal graph G, if there is an unobservable node U_i which is a parent for both of them, then path $V_1 \leftarrow U_i \rightarrow V_2$ is called a *bidirected link*[3].

A path between V_1 and V_2 is called an *extended bidirected link* (or *divergent path*) if (i) there is at last one internal node in that path; (ii) all the internal nodes in the path are unobservable nodes; (iii) one and only one internal node in the path is a divergent node and there is no convergent internal node. In other words, an extended bidirected link between V_1 and V_2 means that for V_1 and V_2 there is an unobservable node U_i, such that there are two directed paths from U_i to V_1 and from U_i to V_2 respectively, and all nodes in these two paths are unobservable.

In a Bayesian network with hidden variables, if each hidden variable is a root node with exactly two observed children, then corresponding model is called a *semi-Markovian model*.

We now introduce a way of reducing the size of causal graphs that preserves the answer to an identifiability question. It is more convenient to work with the reduced graphs than with the original, larger ones. Studying definition (4) and (5), we note that, if there is an unobservable variable in graph G that has no child, then it can be summed out in both (4) and (5) and removed. Formally, If we have a model $M =< N, U, G_{N \cup U}, P(v_i|pa(V_i)) >$, $U' \in U$ and U' has no child in $G_{N \cup U}$, then the identification problem in M is equal to the identification problem in $M' =< N, U \backslash \{U'\}, G', P'(v_i|pa_i) >$, where G' is the subgraph of $G_{N \cup U}$ obtained by removing node U' and all links attached with it. $P'(v_i|pa(V_i))$ is obtained by removing all $P(u'|pa(U'))$ in set $P(v_i|pa(V_i))$. The overall distribution (of all variables common to both models) and the causal distribution (of only the observable variables) in these two models are still the same.

By repeating the transformation given above, any general causal model can be transformed to a model in which each unobservable variable is an ancestor of one or more observable variables. (This is analogous to barren node removal

[3]We use this term, because in all cases, the three-node structure can be replaced by the two observable nodes with a special *bidirected* edge between them.

in Bayesian networks.) From now on in this paper, we assume that all models we study satisfy this property.

We conclude this section by giving several simple graphical definitions that will be needed later. For a given set of variables C, we define *directed observable parent set* $DOP(C)$ as below. A node V belongs to $DOP(C)$ if and only if both of these two conditions are satisfied: i) V is an observable node; ii) there is a directed path from V to an element of C such all the internal nodes on that path are observable nodes. We define the *directed unobservable parent set* $DUP(C)$ by replacing the word "observable" with "unobservable" in the previous definition.

For a given observable variable set $C \subseteq N$, let G_C denotes the subgraph of G composed only of variables in $C \cup DUP(C)$ and all the links between variable pairs in $C \cup DUP(C)$. Let $An(C)$ be the union of C and the set of ancestors of the variables in C and $De(C)$ be the union of C and the set of descendants of the variables in C. An observable variable set $S \subseteq N$ in graph G is called an *ancestral set* if it contains all its own observed ancestors, i.e., $S = An(S) \cap N$.

3 Lemmas

In a recent paper [3], Tian and Pearl propose an algorithm to answer the identifiability question in semi-Markovian models. Huang and Valtorta [4] proved its completeness for semi-Markovian models. We will extend the algorithm to general causal graphs in the next two sections and prove that the algorithm is complete.

In this section we present some lemmas that will be used in the next two sections. We begin with two lemmas proved in [2]. Since our definition of $Q[S]$ is equal to the definition of $Q[S]$ in [2], Lemma 1 in [2] is still correct, and therefore we have:

Lemma 1 *Let $W \subseteq C \subseteq N$. If W is an ancestral set in G_C, then*

$$\sum_{V_i \in C \setminus W} Q[C] = Q[W] \tag{10}$$

We recall that G_C includes all variables in C and the subset of the unobservable variables in G for which there is a path to a variable in C, and such that all the internal nodes (if they are exist) in that path are in U. The lemma says that in such a subgraph, if W is a set of observable variables whose ancestor set includes no other observable variables in the subgraph, then $Q[W]$ can be calculated directly from $Q[C]$ by marginalizing variables in $C \setminus W$. In particular, note that if $Q[C]$ is identifiable, then $Q[W]$ is also identifiable. We will exploit this observation later on.

Another very important lemma is also from [2]. We only use the first two parts of it, which are:

Lemma 2 *Let $H \subseteq N$, and we have c-components $H'_1, \ldots, H'_n$ in the sub graph G_H, $H_i = H'_i \cap H$, $1 \leqslant i \leqslant n$, then*

(i) $Q[H]$ can be decomposed as

$$Q[H] = \prod_{i=1}^{n} Q[H_i] \tag{11}$$

(ii) Each $Q[H_i]$ is computable from $Q[H]$, in the following way. Let k be the number of variables in H, and let a topological order of variables in H be $V_{h_1} < \ldots < V_{h_k}$ in G_H, Let $H^{(j)} = \{V_{h_1}, \ldots, V_{h_j}\}$ be the set of variables in H ordered before V_{h_j} (including V_{h_j}), $j = 1, \ldots, k$,and $H^{(0)} = \phi$. Then each $Q[H_i]$,$i = 1, \ldots, n$,is given by

$$Q[H_i] = \prod_{\{j|V_{h_j} \in H_i\}} \frac{Q[H^{(j)}]}{Q[H^{(j-1)}]} \tag{12}$$

where each $Q[H^{(j)}]$, $j = 0, 1, \ldots, k$, is given by

$$Q[H^{(j)}] = \sum_{h \backslash h^{(j)}} Q[H] \tag{13}$$

Lemma 2 means that if $Q[H]$ is identifiable, then each $Q[H_i]$ is also identifiable.

In the special case for which $H = N$, Lemma 2 implies that, for a given graph G, because $Q[N]$ is identifiable, $Q[C \cap N]$ is identifiable for each c-components C in G.

Lemma 3 *Let $S, T \subset N$ be two disjoint sets of observable variables, If $P_T(S)$ is not identifiable in G, then $P_T(S)$ is not identifiable in the graph resulted from adding a directed or bidirected edge to G. Equivalently, if $P_T(S)$ is identifiable in G, then $P_T(S)$ is still identifiable in the graph resulted from removing a directed or bidirected edge from G.*

Intuitively, this lemma says the unidentifiablility would not change by adding any links. This property is mentioned in [6]. A formal prove of it in semi-Markovian model can be found in [3]. We give out a proof of this lemma in appendix A, which is straightly follow the proof in [3].

Lemma 4 *Let $S, T \subset N$ be two disjoint sets of observable variables, If S_1 and T_1 are subset of S, T, and $P_{T_1}(S_1)$is not identifiable in a subgraph of G, which does not include node in $S \backslash S_1 \cup T \backslash T_1$, then $P_T(S)$ is not identifiable in the graph G.*

Prove: Assume that $P_{T_1}(S_1)$is not identifiable in a subgraph of G, which we will name G' , and which does not include node $S \backslash S_1 \cup T \backslash T_1$. we can add all node in G but do not in G' into G' as isolated nodes. Then we have(trivially) that $P_T(S)$ is not identifiable in this new graph. According to lemma 3, it is not identifiable in graph G either. □

Lemma 5 *Let $A \subset B \subset N$. $Q[A]$ is computable from $Q[B]$ if and only if $Q[A]_{G_B}$ is computable from $Q[B]_{G_B}$*

[3] gives a proof of this lemma when the models are semi-Markovian. Note that $Q[A] = P_{V \setminus A}(A)$, the only if part of this lemma can be gotten directly from lemma 4. A formal proof of the if part can be found in Appendix B.

4 Identify Algorithm For $Q[S]$

Based the lemmas in the last section, we give out an algorithm to calculate $Q[S]$. Here $S \subset N$ is a subset of observable variables.

The first version of this algorithm is given by Tian and Pearl [3] on semi-Markovian models. They pointed out, in [3], that it could be transfered to general models. Huang and Valtorta [4] have proved Tian and Pearl's algorithm is sound and complete on semi-Markovian models. The algorithm below is the transfered version on general models and we also give out a proof of its soundness and completeness in this section.

Assume $N(G)$ be partitioned into $N_1, \ldots, N_k$ in G, each of them belongs to a c-components, and we have c-components $S'_1, \ldots, S'_l$ in graph G_S, $S_j = S'_j \cap S$, $1 \leqslant j \leqslant l$.

Based on theorem 2, for any model on graph G, We have

$$Q[S] = \prod_{j=1}^{l} Q[S_j] \tag{14}$$

Because each S_j,$j = 1, \ldots, l$, is a c-component in G_S, which is a subgraph of G, it must be included in one N_j, $N_j \in \{N_1, \ldots, N_k\}$. We have:

Lemma 6 *$Q[S]$ is identifiable if and only if each $Q[S_j]$ is identifiable in graph G_{N_j}.*

Prove: only if part: based on lemma 5, each $Q[S_j]$ is identifiable in graph G_{N_j} means each $Q[S_j]$ is identifiable from $Q[N_j]$ on graph G. When we have $Q[N]$, according to lemma 2, we can compute all the $Q[N_j]$s. So, each $Q[S_j]$ is identifiable from $Q[N]$. Based on equation 14, $Q[S]$ is identifiable.

If part: If one $Q[S_j]$ is unidentifiable in $Q[N_j]$ in graph G_{N_j}, then from lemma 4, we have $Q[S]$ is unidentifiable. □

Now let us consider how to compute $Q[S_j]$ from $Q[N_j]$. Note that $S_j \subset N_j$ and both G_{N_j} and G_{S_j} are graphs with just one c-component.

Let $F = An(S_j)_{G_{N_j}} \cap N_j$

If $F = S_j$, that is, S_j is an ancestral set in G_{N_j}, the by lemma 1, $Q[S_j]$ is computable as: $Q[S_j] = \sum_{N_j \setminus S_j} Q[N_j]$.

If $F = N_j$, we will prove $Q[S_j]$ is not identifiable in G_{N_j}.

If $S_j \subset F \subset N_j$, by lemma 1, we know $Q[F] = \sum_{N_j \setminus F} Q[N_j]$.

Assume that in graph G_F, S_j is contained in a c-component H', Assume $H = H' \cap N$,(Note that here S_j must belong to one c-component). By lemma 2, $Q[H]$ is computable from $Q[F]$, and is computable with Equation 12.

We obtain that the problem of whether $Q[S_j]$ is computable from $Q[N_j]$ is reduced to that of whether $Q[S_j]$ is computable from $Q[H]$.

Using lemma 5, we know $Q[S_j]$ is computable from $Q[N_j]$ in G_{N_j} if and only if $Q[S_j]$ is identifiable form $Q[H]$ in graph G_H.

Next, we give out the algorithm (which follows [3]) to get $Q[C]$ form $Q[T]$.

Function Identify (C,T,Q**)**

INPUT: $C \subseteq T \subseteq N$, $Q = Q[T]$, G_T *and* G_C *are both composed of one single c-component.*

OUTPUT: Expression for $Q[C]$ *in terms of Q or FAIL.*

Let $A = An(C)_{G_T} \cap T$

i) If $A = C$, *output* $Q[C] = \sum_{T\setminus C} Q[T]$.

ii) If $A = T$, *output FAIL.*

iii) If $C \subset A \subset T$

1. *Assume that in* G_A, C *is contained in a c-component* T_1', $T_1 = T_1' \cap A$.
2. *Compute* $Q[T_1]$ *from* $Q[A] = \sum_{T\setminus A} Q[T]$ *with lemma 2*
3. *Output Identify(*C,T_1,$Q[T_1]$*).*

From the discussions above, we know i) and iii) always work. ii) is handled by lemma below.

Lemma 7 *In a general Markovian model* G, *if*

1. G *itself is a c-component.*
2. $S \subset N(G)$ *and* G_S *has only one c-component.*
3. *All variables in* $N\setminus S$ *are ancestors of* S.

then $Q[S]$ *is unidentifiable in* G.

The proof of this lemma is in appendix C. Putting all the analysis above together, we have

Algorithm Computing $Q[S]$

INPUT: $S \subseteq N$.

OUTPUT: Expression for $Q[S]$ *or FAIL.*

Let $N(G)$ *be partitioned into* $N_1, \ldots, N_k$, *each of them belonging to a c-components in* G, *and* S *be partitioned into* $S_1, \ldots, S_l$, *each of them belonging to a c-components in* G_S, *and* $S_j \subseteq N_j$. *We can*

i), Compute each $Q[N_j]$ *with lemma 2.*

ii), Compute each $Q[S_j]$ *with Identify algorithm above with* $C = S_j$,$T = N_j$,$Q = Q[N_j]$.

iii), If in ii), we get Fail as return value of Identify algorithm of any S_j, *then* $Q[S]$ *is unidentifiable in graph G; else* $Q[S]$ *is identifiable and* $Q[S] = \prod_{j=1}^{l} Q[S_j]$

Theorem 1 *The computing* $Q[S]$ *algorithm is sound and complete.*

From theorem 1, we can have the two lemmas below,

Lemma 8 *If $S \subset N$ in graph G and e is a link that gets out of one S node, graph G' is the same as graph G except it does not have link e. Then $Q[S]$ is identifiable in graph G if and only if $Q[S]$ is identifiable in graph G'.*

Proof: Note that e is a link that gets out of an S node. So, graph G and G' have the same c-component partition. Any c-component in G is also a c-component in G', and vice versa. Graph G_S and G'_S also have the same c-component partition. Any c-component in G_S is also a c-component in G'_S, and vice versa.

When we use the above algorithm to compute $Q^G[S]$ in causal graph G and compute $Q^{G'}[S]$ in causal graph G', the only difference is between computing $Q^G[S_j]$ and $Q^{G'}[S_j]$ with Identify algorithm above with $C = S_j$,$T = N_j$,$Q = Q[N_j]$, where link e is out of a node in S_j.

We note that when computing $Q[S_j]$ with the Identify algorithm, in both case we get the same A and T_1, and the same c-component partition.

From Identify function, computing $Q[S]$ algorithm and theorem 1, we know $Q[S]$ is identifiable in graph G if and only if $Q[S]$ is identifiable in graph G'. □

We also have

Lemma 9 *Let $S \subset N$ in graph G and graph G' be obtained by removing all links getting out from S nodes in graph G. Then $Q[S]$ is identifiable in graph G if and only if $Q[S]$ is identifiable in graph G'.*

Proof: This result directly follows from lemma 8 above. □

5 Identify Algorithm For $P_t(s)$

Lemma 10 *Assume $S \subset N$ in graph G, $X_1 \in S$, $X_2 \in S$. Let $< X_1, U_1, \ldots, U_k, X_2 >$ be a directed path from X_1 to X_2 in G, with $U_i \in U(G)$, $1 \leqslant i \leqslant k$, and let $T \subset N$ and $T \cap S = \phi$. Let graph G' be obtained by removing link $< X_1, U_1 >$ from graph G. If $P_T(S)$ is unidentifiable in graph G', then $P_T(S\backslash\{X_1\})$ is unidentifiable in G.*

The proof of this lemma is in Appendix D.

We define *s-ancestor set* of $S \subseteq N$ in graph G here. The *s-ancestor set* D of S in G is a observable variable set that $S \subseteq D \subseteq N$ and $D = An(S)$ in G_D.

Lemma 11 *Assume D is an s-ancestor set of observable node set S on graph G, then $\sum_{D\backslash S} Q[D]$ is identifiable if and only if $Q[D]$ is identifiable.*

Proof: if part is easy, by definition, if $Q[D]$ is identifiable, $\sum_{D\backslash S} Q[D]$ is identifiable.

If $Q[D]$ is unidentifiable, then we know form the lemma 9 that $Q[D]$ is unidentifiable in graph G', here G' is gotten by removing all links that get out from nodes in D from G.

Because D is a s-ancestor set of S, we can find an order of nodes in $D\backslash S$ as $X_1, \ldots, X_k$, and in graph G for each X_i, $1 \leqslant i \leqslant k$, there is a directed path from X_i to one node in $S \cup \{X_1, \ldots, X_{i-1}\}$, and all nodes in the middle of that path are unobservable. Assume for a given X_i, $1 \leqslant i \leqslant k$, the link, which gets out from X_i in G but is not exist in G', is e_i. And graph G_i is gotten by add link e_i to graph G_{i-1}, $G_0 = G'$.

Notes we have $Q[D] = P_{N\backslash D}(D)$ is unidentifiable in G'. From lemma 10, we have $P_{N\backslash D}(D\backslash\{X_1\})$ is unidentifiable in graph G_1, using this lemma again, we have $P_{N\backslash D}(D\backslash\{X_1, X_2\})$ is unidentifiable in graph G_2, and finally, we have $P_{N\backslash D}(S)$ is unidentifiable in graph G_k. Notes G_k is a subgraph of G, according to lemma 3, if $P_{N\backslash D}(S)$, which equals to $\sum_{D\backslash S} Q[D]$, is unidentifiable in G_k then it is unidentifiable in G. □

Based the lemmas above, we get an algorithm to solve the identifiability problem on general Markovian models.

What we want to compute is:

$$P_t(s) = \sum_{N\backslash(T\cup S)} P_t(n\backslash t) = \sum_{N\backslash(T\cup S)} Q[N\backslash T] \tag{15}$$

Let $D = An(S)_{G_{N\backslash T}}$. D is an ancestral set in graph $G_{N\backslash T}$, Lemma 1 allows us to conclude that $\sum_{N\backslash(T\cup D)} Q[N\backslash T] = Q[D]$. Therefore, we can rewrite $P_t(s)$ from Equation (15) as:

$$P_t(s) = \sum_{N\backslash(T\cup S)} Q[N\backslash T] = \sum_{D\backslash S} \sum_{N\backslash(T\cup D)} Q[N\backslash T] = \sum_{D\backslash S} Q[D] \tag{16}$$

Note that D is a s-ancestor set of S, according to lemma 11, $\sum_{D\backslash S} Q[D]$ is identifiable if and only if $Q[D]$ is identifiable.

Now the identifiability problem of $P_T(S)$ is transfered to the identifiability problem of $Q[D]$, which can be solved by the algorithm in the last section.

Finally, we give out the algorithm which follows [3] to deal the identifiable problem.

Algorithm Computing $P_T(S)$
INPUT: two disjoint observable variable sets $S, T \subset N$.
OUTPUT: the expression for $P_T(S)$ or FAIL.

1. *Let $D = An(S)_{G_{N\backslash T}}$*
2. *Using the Computing $Q[S]$ algorithm in last section to compute $Q[D]$.*
3. *If the algorithm returns FAIL, then output FAIL.*
4. *Else, output $P_T(S) = \sum_{D\backslash S} Q[D]$*

Our discussion above shows,

Theorem 2 *The above computing $P_T(S)$ algorithm is sound and complete.*

6 Conclusion

In this paper, we review the identify algorithm given by J.Tian and J.Pearl, which can be used only on semi Markovian graphs. We extend that algorithm into an identify algorithm that can be used on general causal graphs and prove that the extended algorithm is sound and complete. This result shows the power of the algebraic approach to solving identifiability problems and closes the identifiability problem.

Future work includes implementing the modified identify algorithm and analyzing its efficiency, extending the results of this paper to conditional causal effects, and providing an explanation of the causal effect formula found by the identify algorithm in terms of applications of the rules of the do calculus by J.Pearl in [6].

References

[1] J. Tian and J. Pearl, "A general identification condition for causal effects," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pp. 567–573, 2002.

[2] J. Tian and J. Pearl, "On the testable implications of causal models with hidden variables," in *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence(UAI-02)*, pp. 519–527, 2002.

[3] J. Tian and J. Pearl, "On the identification of causal effects, Technical report 290-L," tech. rep., Cognitive Systems Laboratory, University of California at Los Angeles, 2003. Extended version available at http://www.cs.iastate.edu/ jtian/r290-L.pdf.

[4] Y. Huang and M. Vartolta, "On the completeness of an identifiability algorithm for semi-markovian models, TR-2006-001," tech. rep., University of South Carolina Department of Computer Science, 1 2006. available at http://www.cse.sc.edu/ mgv/reports/tr2006-001.pdf.

[5] J. Pearl, "Causal diagrams for empirical research," *Biometrika*, vol. 82, pp. 669–710, 12 1995.

[6] J.Pearl, *Causality: Models, Reasoning,and Inference*. New York,USA: Cambridge University Press, 2000.

[7] D. Galles and J. Pearl, "Testing identifiability of causal effects," in *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence(UAI-95)*, pp. 185–195, 1995.

[8] J. Pearl and J.M.Robins, "Probabilistic evaluation of sequential plans from causal models with hidden variables," in *Proceedings of the Eleveth Annual Conference on Uncertainty in Artificial Intelligence(UAI-95)*, pp. 444–453, 1995.

[9] M.Kuroki and M. Miyakawa, "Identifiability criteria for causal effects of joint interventions," *Journal of the Japan Statistical Society*, vol. 29(2), pp. 105–117, 1999.

[10] J. Robins, "Causal inference from complex longitudinal data," in *Latent Variable Modeling with Applications to Causality, Volume 120 of Lecture Notes in Statistics* (M. Berkane, ed.), pp. 69–117, New York: SpringerVerlag, 1997.

[11] J. Tian, "Identifying conditional causal effects," in *Proceedings of the Twentieth Annual Conference on Uncertainty in Artificial Intelligence(UAI-04)*, pp. 561–568, 2004.

[12] I. Shpitser and J. Pearl, "Identification of joint interventional distributions in recursive semi-markovian causal models, Technical report R_327," tech. rep., Cognitive Systems Laboratory, University of California at Los Angeles, 2006. available at http://ftp.cs.ucla.edu/pub/stat_ser/r327.pdf.

[13] S. Lauritzen, "Causal inference from graphical models," 2001.

[14] T. Verma, "Graphical aspects of causal models, Technical Report, R-191," tech. rep., Cognitive Systems Laboratory, University of California at Los Angeles, 1993.

Appendix A

This appendix contains the proof of lemma 3, which we repeat:

Lemma 3 Let $S, T \subset N$ be two disjoint sets of observable variables, If $P_T(S)$ is not identifiable in G, then $P_T(S)$ is not identifiable in the graph resulting from adding a directed or bidirected edge to G. Equivalently, if $P_T(S)$ is identifiable in G, then $P_T(S)$ is still identifiable in the graph resulted from removing a directed or bidirected edge from G.

Proof: This proof follows [3]. The only difference is that in [3] the models are semi Markovian, while here we deal with general casual networks.

If $P_T(S)$ is not identifiable in G, then there exist two models with the same causal graph G, M_1 and M_2, such that for all instances $N = n$

$$P^{M_1}(n) = P^{M_2}(n) > 0 \tag{17}$$

but for at least one instance $T = t, S = s$

$$P_t^{M_1}(s) \neq P_t^{M_2}(s) \tag{18}$$

where:

$$P(n) = \sum_{U_k \in U} \prod_{V_i \in N} P(v_i|pa(V_i)) \prod_{V_j \in U} P(v_j|pa(V_j)) \tag{19}$$

$$P_t(s) = \sum_{V_l \in (N \setminus S) \setminus T} \sum_{U_k \in U} \prod_{V_i \in N \setminus T} P(v_i|pa(V_i)) \prod_{V_j \in U} P(v_j|pa(V_j)) \tag{20}$$

For a graph G' with extra edges added to G, we will construct new models based on M_1 and M_2 in which the added edges are ineffective.

Let G' be the graph identical to G except having an extra edge $X \longrightarrow V_j$.

We construct two models M_1' and M_2' with the causal graph G' as

$$P^{M_k'}(v_i|pa(V_i)) = P^{M_k}(v_i|pa(V_i)), i \neq j, k = 1, 2 \tag{21}$$

$$P^{M_k'}(v_j|pa(V_j), X) = P^{M_k}(v_j|pa(V_j)), k = 1, 2 \tag{22}$$

here, $pa(V_i)$ is an instance of $Pa(V_i)$, the parent set of V_i in graph G.

When (M_1, M_2) satisfies 17 and 18, (M_1', M_2') satisfies them also. So, $P_T(S)$ is not identifiable in G'. $\square$

Appendix B

This appendix contains the proof of lemma 5, which we repeat:

Lemma 5 Let $A \subset B \subset N$. $Q[A]$ is computable from $Q[B]$ if and only if $Q[A]_{G_B}$ is computable from $Q[B]_{G_B}$

Proof: Note that $Q[A] = P_{V \setminus A}(A)$. The only if part of this lemma follows from lemma 4. Tian and Pearl [3] give a proof of this statement when models are semi-Markovian. Our proof follows their proof on the if part.

Assume that $Q[A]$ is not computable from $Q[B]$. Then, there exist two models, M_1 and M_2, with the same causal graph G, that satisfy, for any (t, a, c),

$$Q^{M_k}[B](b,t) = P^{M_k}_{N \setminus B}(B)(b,t) = \sum_{U_k \in U} \prod_{V_i \in \{B \cup U\}} P(v_i|pa'(V_i), t_i, u^i), k = 1, 2 \tag{23}$$

where t is an instance of $T = N \setminus B$, $Pa'(V_i) = Pa(V_i) \cap B$, $T_i = Pa(V_i) \cap T$, $U^i = Pa(V_i) \cap U$. We also have

$$Q^{M_1}[B](b,t) = Q^{M_2}[B](b,t) > 0 \tag{24}$$

for all values (b, t), but

$$Q^{M_1}[A](b', t') \neq Q^{M_2}[A](b', t') \tag{25}$$

for a particular value (b', t').

We construct two models, M_1' and M_2' with the same causal graph G_B as

$$P^{M_k'}(v_i|pa'(V_i), u^i) = P^{M_k}(v_i|pa'(V_i), T_i = t_i', u^i), k = 1, 2 \tag{26}$$

Note that $Q[B]_{G_B}$ can be written as:

$$\begin{aligned} Q[B]^{M_k'}_{G_B}(b) &= \textstyle\sum_{U_k \in DUP(B)} \prod_{V_i \in \{B \cup DUP(B)\}} P^{M_k}(v_i|pa'(V_i), t_i, u^i) \\ &= \textstyle\sum_{U_k \in U} \prod_{V_i \in \{B \cup U)\}} P^{M_k}(v_i|pa'(V_i), t_i, u^i), k = 1, 2 \end{aligned} \tag{27}$$

This is because the nodes in $U \setminus DUP(B)$ have no effect on B nodes when the T nodes are set.

Then we have

$$Q[B]^{M'_k}_{G_B}(b) = Q[B]^{M_k}(b, t'), Q[A]^{M'_k}_{G_B}(b) = Q[A]^{M_k}(b, t'), k = 1, 2 \tag{28}$$

From the discussion above, we have:

$$Q^{M'_1}[B]_{G_B}(b) = Q^{M'_2}[B]_{G_B}(b) > 0 \tag{29}$$

for all values b, but

$$Q^{M'_1}[A]_{G_B}(b') \neq Q^{M'_2}[A]_{G_B}(b') \tag{30}$$

for value b', which means that $Q[A]_{G_B}$ is not computable from $Q[B]_{G_B}$. □

Appendix C

This appendix contains the proof of lemma 7, which we repeat:
lemma 7 In a (general) Markovian model G, if

1. G itself is a c-component.
2. $S \subset N(G)$ and G_S has only one c-component.
3. All variables in $N \backslash S$ are ancestors of S.

then $Q[S]$ is unidentifiable in G.

We call these three properties *unidentifiable properties*.

In this section, we first introduce the projection process to transfer general Markovian models to semi-Markovian models; then we prove if the projection, which is a semi-Markovian model, is unidentifiable then the original graph is unidentifiable. Finally we show any semi-Markovian model we are studying in this section is unidentifiable.

Projection

Verma [14] showed any Bayesian network with arbitrary hidden variables can be converted to a semi-Markovian model by constructing its projection.

Projection *The projection of a DAG G over observable nodes set N and unobservable node set U, denoted by $PJ(G, N)$, is a DAG over N with bidirected edges constructed as follows:*

1. Add each variable in N as a node of $PJ(G, N)$.

2. For each pair of variables $X, Y \in N$, if there is an edge between them in G, add the edge to $PJ(G, N)$.

3. For each pair of variables $X, Y \in N$, if there exists a directed path from X to Y in G such that every internal node on the path is in U, add edge $X \to Y$ to $PJ(G, N)$ (if it does not exist yet).

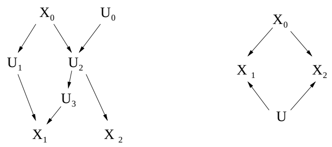

Figure 1: Semi-Markovian Projection

4. For each pair of variables $X, Y \in N$, if there exists a divergent path between X and Y in G such that every internal node on the path is in U, add a bidirected edge between X and Y in $PJ(G, N)$.

Figure 1 is an example of a general Markovian model and its semi-Markovian projection.

Verma [14] shows that G and $PJ(G, N)$ have the same set of conditional independence relations among N. Tian and Pearl [2] shows G and $PJ(G, N)$ share additional non-independence equality constraints among variables. Our result does not depend on these observation.

From the definition of the projection, we have

Lemma 12 *Assume model G and $S \in N(G)$ satisfy the three unidentifiable properties. Then, $PJ(G, N(G))$ and S all satisfy the three unidentifiable properties.*

Proof: First note that for any two observable nodes in G, if they are in the same c-component, then they are still in the same c-component in $PJ(G, N(G))$.

Second, for two observable node X and Y, if X is an ancestor of Y in G, then X is still Y's ancestor in $PJ(G, N(G))$. $\Box$

From Projection to Original Model

A property of projection $PJ(G, N)$ is given in the following:

Lemma 13 *If $Q[S]$ is unidentifiable in $PJ(G, N)$ then $Q[S]$ is unidentifiable in G.*

Prove: Assume we have graph G, $V(G) = N \cup U$, $S \subset N$.

$Q[S]$ is unidentifiable in $PJ(G, N)$ means we have two model M_1 and M_2 on graph $PJ(G, N)$ that

$$\begin{array}{c} \sum_{U \in U(PJ(G,N))} \prod_{V \in V(PJ(G,N))} P^{M_1}(V|Pa(V)) = \\ \sum_{U \in U(PJ(G,N))} \prod_{V \in V(PJ(G,N))} P^{M_2}(V|Pa(V)) > 0 \end{array} \tag{31}$$

but there exists an instance $N = n$, for which

$$\begin{array}{c} \sum_{U \in U(PJ(G,N))} \prod_{V \in (U(PJ(G,N)) \cup S)} P^{M_1}(V|Pa(V_i))(n) \neq \\ \sum_{U \in U(PJ(G,N))} \prod_{V \in (U(PJ(G,N)) \cup S)} P^{M_2}(V|Pa(V_i))(n) \end{array} \tag{32}$$

We assume the state space for each node V_i in $PJ(G, N)$ is $S(V_i)$

Now based on M_1 and M_2, we construct two models M'_1 and M'_2 on a subgraph of G. We define a state space set $SS(X)$ for each node X in $V(G)$, and at the beginning we set $SS(X) = \phi$. Then we follow the three steps below.

A) For each node X in N, we add its state space in $PJ(G, N)$ to its state space set. That is $SS(X) = \{S(X)\}$.

B) If in $PJ(G, N)$, observable node X is a parent of observable node Y, from the projection precess, we know there are some directed paths from X to Y in G such that all internal nodes on those paths are in U. We select one of these paths and add state space $S(X)$ into the state space sets of all the internal nodes on that path if it is not in them yet.

C) For any bidirected link in $PJ(G, N)$, assume it is between observable nodes X, Y and the unobservable node on the link is U_{xy}. From the projection process, we know there exists at least one divergent path between X and Y in G such that every internal node on the path is in $U(G)$. We select the shortest path that satisfies this property and denote the unobservable node with divergent links on that path as U'_{xy}. Then we add the state space of U_{xy} to the state space set of all nodes on that path if it is not in them yet, except X and Y.

After the three steps above, we remove from G all unobservable nodes whose state space sets are still empty. The resulting graph G' is a subgraph of G. From lemma 4, we know that if $Q[S]$ is unidentifiable in G', then it is unidentifiable in G. Our model construction below is on G'.

For any observable node X in $PJ(G, N)$, we note the set of all X's parents' state space as $SPa(X)$. If Y is one of X's parents in $PJ(G, N)$, then there is at least one parent node Z of X in G', such that state space $S(Y)$ is in Z's state space set.

We define the state space of each node in G' as the product of its state space set.

Let us now consider the conditional probalitity table of each node X in G' as a mapping from the product of $Pa(X)$'s state space to that X's state space.

Based on our construction, the product of $Pa(X)$'s state space can be transformed to the product of all state spaces in a bag that consists of all the state space sets of nodes in $Pa(X)$. We call this bag $PB(X)$, which is $\sum_{Y \in Pa(X)} SS(Y)$. The CPT of X maps the product of $PB(X)$ to X's state space.

If X is an observable node, then its CPT in $PJ(G, N)$ is defined as a map from the product of $SPa(X)$, that is $\prod_{Y \in SPa(X)} S(Y)$, to $S(X)$. Note that $SPa(X)$ is a subset of $PB(X)$. We define for $k = 1, 2$,

$$\begin{aligned} &P^{M'_k}(X = x | SPa(X) = a, (PB(X) - SPa(X)) = b) = \\ &P^{M_k}(X = x | SPa(X) = a) \end{aligned} \tag{33}$$

If the same node state space in $SPa(X)$ appears more than once on $PB(X)$, then we just arbitrarily select one of them in the above definition.

If X is an unobservable node in G', assume its state space set $SS(X) = \{Y_1, \ldots, Y_n, Z_1, \ldots, Z_m\}$, where Y_i, $1 \leqslant i \leqslant n$, are state spaces that also exist in $PB(X)$, while $Z_1, \ldots, Z_m$ do not.Note that $Z_1, \ldots, Z_m$ come from step C) of our construction when X is selected as U'_{xy}.The CPT of X is defined as

$$\begin{array}{l} P^{M'_k}(X=(y_1,\ldots,y_n,z_1,\ldots,z_m)|y'_1,\ldots,y'_n,PB(X)\backslash\{Y'_1,\ldots,Y'_n\}=b) \\ =\begin{cases} \prod_{Z_i\in\{Z_1,\ldots,Z_m\}} P^{M_k}(Z_i=z_i) & all\ y_j=y'_j \\ 0 & exist\ y_j\neq y'_j \end{cases} \end{array} \tag{34}$$

Here $S(Y'_j)$ is the same state space as $S(Y_j)$ in $PB(X)$, y'_j is an instance of it. If a same node state space in $\{Y_1,\ldots,Y_n\}$ appears more than once on $PB(X)$, then we just arbitrarily select one of them in the above definition.

If $PB(X)\backslash\{Y'_1,\ldots,Y'_n\}$ is empty, then $\prod_{Z_i\in\{Z_1,\ldots,Z_m\}} P^{M_k}(Z_i=z_i)$ is 1 in above equation.

Based on the definition of causal effect and the construction above, we have for $i=1,2$ and for any instance v of $V(PJ(G,N))$, which is the whole variable set on $PJ(G,N)$, we can get a mapping instance for all variables in $V(G')$. Note that for each variable in $V(G')$, its state space is the product of some state spaces of variable in $V(PJ(G,N))$. So, the mapping is given by forcing each part of the product to have the same value that the state space has in v. If we call the new instance on $V(G)$ as v', we have $P^{M_k}(v)=P^{M'_k}(v')$. For any instance v'' of $V(G')$ that could not be obtained by this kind of mapping $P^{M'_k}(v'')=0$.

Based on definition, we have the observed probability distribution of M'_i is completely as same as it for M_i, which means for any instance n of $N(G)$. $P^{M_k}(n)=P^{M'_k}(n)>0$.

With the model construction given above, for any instance v of $V(PJ(G,N))$, if $v=(u,t,s)$, where u,t,s are instance of $U, N\backslash S=T$ and S, and we call the new instance on $V(G)$ after mapping as v', we have

$P^{M_k}_{N\backslash S=t}(v)=P^{M'_k}_{N\backslash S=t}(v')$. This is because the causal effect part is obtained by removing the conditional probability table of $S\backslash N$ nodes from the joint probability formula. And that part is the same for M_k and M'_k under our construction.

For any instance v'' of $V(G')$ which cannot be obtained by the mapping, set $P^{M'_k}_{N\backslash S=t}(v'')=0$. This is because for the causal graph, when we remove all links into the $N\backslash S$ nodes, all the CPTs for unobservable nodes are unchanged. So, if an instance is not from the mapping, the inconsistency of values in an unobservable node will cause the whole formula to be zero.

So, the causal effect on S is also completely the same.

This proves the result that if $PJ(G,N)$ is unidentifiable, then G must be unidentifiable.

□

Unidentifiability of Projection

We also have

Lemma 14 *In a semi-Markovian model G, if*

1. G itself is a c-component.

2. $S \subset N(G)$ and G_S has only one c-component.

3. All variables in $N \backslash S$ are ancestors of S.

then $Q[S]$ is unidentifiable in G.

Proof: this lemma is proved by Huang and Valtorta in [4] and by Shpitser and Pearl in [12]. □

Proof of lemma 7

Proof: Compose the three lemmas in this section. □

Appendix D

This appendix contains the proof of lemma 10, which we repeat:

Lemma 10 Assume $S \subset N$ in graph G, $X_1 \in S$, $X_2 \in S$. Let $< X_1, U_1, \ldots, U_k, X_2 >$ be a directed path from X_1 to X_2 in G, with $U_i \in U(G)$, $1 \leqslant i \leqslant k$, and let $T \subset N$ and $T \cap S = \phi$. Let graph G' be obtained by removing link $< X_1, U_1 >$ from graph G. If $P_T(S)$ is unidentifiable in graph G', then $P_T(S \backslash \{X_1\})$ is unidentifiable in G.

Proof: We assume path $X_1 \in S$, $X_2 \in S$, $< X_1, U_1, \ldots, U_k, X_2 >$ is a shortest path. Therefore, we may not remove any U node in this path but keep it still be a path from X_1 to X_2. Otherwise, we can remove some U node in it to get a shortest path.

By definition, in graph G', $P_t(s)$ is given by:

$$P_t(s) = \sum_{V_l \in (N \backslash S) \backslash T} \sum_{U_k \in U} \prod_{V_i \in N \backslash T \cup U} P(v_i | pa(V_i)) \tag{35}$$

In graph G,

$$P_t(s \backslash \{x_1\}) = \sum_{V_l \in (N \backslash S) \backslash T \cup \{X_1\}} \sum_{U_k \in U} \prod_{V_i \in N \backslash T \cup U} P(v_i | pa(V_i)) \tag{36}$$

When $P_T(S)$ is unidentifiable in graph G', we know there are two models M_1 and M_2 on G' that: $P^{M_1}(n) = P^{M_2}(n)$, which means:

$$\sum_{U_k \in U} \prod_{V_i \in V} P^{M_1}(v_i | pa(V_i)) = \sum_{U_k \in U} \prod_{V_i \in V} P^{M_2}(v_i | pa(V_i)) > 0 \tag{37}$$

but for given (s,t),$P_t^{M_1}(s) \neq P_t^{M_2}(s)$.

Now, based on M_1 and M_2, we create models M_1' and M_2' on graph G. First, we define a probability function F. F is defined from $S(X_1)$ to $(0,1)$, where $S(X_1)$ is the state space of X_1 in model M_i, $i = 1, 2$. Let F be such that for any

$a \in S(X_1)$, $P(F(a) = 0) > 0$, $P(F(a) = 1) > 0$ and $P(F(a) = 0) + P(F(a) = 1) = 1$.

For any node X, which is not in $\{U_1, \dots, U_k, X_2\}$, we define for $i = 1, 2$ the state space for X in model M'_k to be just the state space of X in model M_k.

For any node X, which is in $\{U_1, \dots, U_k\}$, we define for $i = 1, 2$ the state space for X in model M'_k to be the product of the state space of X in model M_k and state space $S(X_1)$, Here, $S(X_1)$ is the state space of X_1 in M_k.

The state space of X_2 in M'_k is defined as $S(X_2) \times \{0, 1\}$. Here, $S(X_2)$ is the state space of X_2 in M_k,$k = 1, 2$.

Then we define the CPT for all the nodes.

First for any node X that is not in $\{U_1, \dots, U_k, X_2\}$ and has no parent in $\{U_1, \dots, U_k, X_2\}$, then both its parents' state spaces and its state space are the same as those in M_k. We define

$$P^{M'_i}(x|pa(X)) = P^{M_i}(x|pa(X)) \tag{38}$$

For any node X, that is not in $\{U_1, \dots, U_k, X_2\}$ but have some parent in $\{U_1, \dots, U_k, X_2\}$, then its own state space is the same as in M_k but some of its parents' state spaces are changed. If one of those parent is node Y, the state space of Y becomes $S(Y) \times S(X_1)$ or $S(Y) \times (0, 1)$. We define

$$P^{M'_i}(x|pa'(X), (y_1, x_1), \dots, (y_n, x_1)) = P^{M_i}(x|pa'(X), y_1, \dots, y_n) \tag{39}$$

Here $\{y_1, \dots, y_n\}$ is an instance of $\{Y_1, \dots, Y_n\} = Pa(X) \cap \{U_1, \dots, U_k, X_2\}$ in model M_k and $pa'(X)$ is an instance of $Pa'(X) = Pa(X) \backslash \{Y_1, \dots, Y_n\}$ in model M_k.

For u_1, which is an instance of U_1 in model M_k, and x_1, which is an instance of X_1 in model M_k, $k = 0, 1$ we define

$$P^{M'_i}((u_1, x_1)|pa(U_1), X_1 = x'_1) = \begin{cases} P^{M_i}(u_1|pa(U_1)) & x_1 = x'_1 \\ 0 & x_1 \neq x'_1 \end{cases} \tag{40}$$

Here, $pa(U_1)$ is an instance of $Pa(U_1)$ in model M_k.

For u_i, which is an instance of $U_i \in \{U_2, \dots, U_k\}$ in model M_k, and x_1, which is an instance of X_1 in model M_k, $k = 0, 1$ we define for node U_i

$$P^{M'_i}((u_i, x_1)|pa'(U_i), (u_{i-1}, x'_1)) = \begin{cases} P^{M_i}(u_i|pa'(U_i), u_{i-1}) & x_1 = x'_1 \\ 0 & x_1 \neq x'_1 \end{cases} \tag{41}$$

Here, $pa'(U_i)$ is an instance of $Pa'(U_i) = Pa(U_i) \backslash \{U_{i-1}\}$ in model M_k.

For x_2, which is an instance of X_2 in model M_k $i = 1, 2$,$m = 0, 1$, we define

$$P^{M'_i}((x_2, m)|pa'(X_2), (u_k, x_1)) = P^{M_i}(x_2|pa'(X_2), u_k) \times P(F(x_1) = m) \tag{42}$$

Here, $Pa'(X_2) \cup \{U_k\}$ is the parent set of X_2 in graph G'.

Note that for a given $(pa(X_2), x_1) = (pa'(X_2), u_k, x_1)$, we have

$$\sum_{x_2,m} P^{M_i'}((x_2, m)|pa'(X_2), u_k, x_1) = \sum_{x_2} P^{M_i}(x_2|pa'(X_2), u_k) \times \sum_{m} P(F(x_1) = m) = 1 \quad (43)$$

Note that under this construction, for any U_i, if its value on the X_1 part is not equal to the value of the X_1 part in one of its parents, the CPT entry of U_i will be zero. This means for a given instance of all variables in model M_k', only those where all X_1 part have the same value will enter into the joint probability calculation.

Then for any instance n of N in model M_1' and M_2', if in n,$X_1 = x_1$ and $X_2 = (x_2, m)$, $m = 0, 1$ we have

$$\begin{aligned}
&P^{M_1'}(n) = \\
&\sum_{U_k \in U} \prod_{V_i \in V} P^{M_1'}(v_i|pa(V_i)) = \\
&\sum_{U_k \in U} \prod_{V_i \in V} P^{M_1}(v_i|pa(V_i))(n) \times P(F(x_1) = m) = \\
&\sum_{U_k \in U} \prod_{V_i \in V} P^{M_2}(v_i|pa(V_i))(n) \times P(F(x_1) = m) = \\
&\sum_{U_k \in U} \prod_{V_i \in V} P^{M_2'}(v_i|pa(V_i)) = \\
&P^{M_2'}(n) > 0
\end{aligned} \quad (44)$$

We know that for given (s, t),$P_t^{M_1}(s) \neq P_t^{M_2}(s)$ and we assume that for that s, $X_1 = x_1$ and $X_2 = x_2$.

Note that $\sum_{X_1} P_t^{M_i}(s \backslash \{x_1\}) \leqslant 1$. This is because after setting the values of T nodes, the result model is still a Bayesian network.

Assume $P_t^{M_1}(s) = a > P_t^{M_2}(s) = b > 0$. If we define $P(F(x_1) = 0) = 0.5$, but $P(F(x) = 0) = (a - b)/4$ for all $x \in S(X_1)$,$x \neq x_1$. We have for $(s \backslash \{x_2\}, (x_2, 0), t)$

$$\begin{aligned}
&P_t^{M_1'}(s \backslash \{x_1\})(S \backslash \{X_1\} = (s \backslash \{x_1, x_2\}, (x_2, 0)), T = t) = \\
&\sum_{V_l \in (N \backslash S) \backslash T \cup \{X_1\}} \sum_{U_k \in U} \prod_{V_i \in V \backslash T} P^{M_1'}(v_i|pa(V_i)) \\
&(S \backslash \{X_1\} = (s \backslash \{x_1, x_2\}, (x_2, 0)), T = t) > \\
&\sum_{X_1 = x_1} \sum_{V_l \in (N \backslash S) \backslash T} \sum_{U_k \in U} \prod_{V_i \in V \backslash T} P^{M_1'}(v_i|pa(V_i)) \\
&(S \backslash \{X_1\} = (s \backslash \{x_1, x_2\}, (x_2, 0)), T = t) = \\
&\sum_{V_l \in (N \backslash S) \backslash T} \sum_{U_k \in U} \prod_{V_i \in V \backslash T} P^{M_1}(v_i|pa(V_i))(S = s, T = t) \times P(F(x_1) = 0) = \\
&= 0.5a
\end{aligned} \quad (45)$$

but,

$$
\begin{aligned}
&P_t^{M_2'}(s\backslash\{x_1\})(S\backslash\{X_1\} = (s\backslash\{x_1, x_2\}, (x_2, 0)), T = t) = \\
&\sum_{V_l \in (N\backslash S)\backslash T \cup \{X_1\}} \sum_{U_k \in U} \prod_{V_i \in V\backslash T} P^{M_2'}(v_i|pa(V_i)) \\
&(S\backslash\{X_1\} = (s\backslash\{x_1, x_2\}, (x_2, 0)), T = t) = \\
&\sum_{X_1 = x_1} \sum_{V_l \in (N\backslash S)\backslash T} \sum_{U_k \in U} \prod_{V_i \in V\backslash T} P^{M_2'}(v_i|pa(V_i)) \\
&(S\backslash\{X_1\} = (s\backslash\{x_1, x_2\}, (x_2, 0)), T = t) + \\
&\sum_{X_1 \neq x_1} \sum_{V_l \in (N\backslash S)\backslash T} \sum_{U_k \in U} \prod_{V_i \in V\backslash T} P^{M_2'}(v_i|pa(V_i)) \\
&(S\backslash\{X_1\} = (s\backslash\{x_1, x_2\}, (x_2, 0)), T = t) < \\
&\sum_{V_l \in (N\backslash S)\backslash T} \sum_{U_k \in U} \prod_{V_i \in V\backslash T} P^{M_2}(v_i|pa(V_i))(S = s, T = t) \times P(F(x_1) = 0) + \\
&\sum_{V_l \in (N\backslash S)\backslash T \cup \{X_1\}} \sum_{U_k \in U} \prod_{V_i \in V\backslash T} P^{M_2}(v_i|pa(V_i)) \\
&(S\backslash\{X_1\} = s\backslash\{x_1\}, T = t) \times P(F(X_i \neq x_1) = 0) \leqslant \\
&0.5b + \sum_{X_1} P_t^{M_2}(s\backslash\{x_1\}) \times (a - b)/4 \leqslant \\
&0.5b + (a - b)/4 < 0.5a
\end{aligned}
\tag{46}
$$

From models M_1' and M_2', we know $P_T(S\backslash\{X_1\}$ is unidentifiable in G.
□