Identifiability in Causal Bayesian Networks: A Sound and Complete Algorithm

Yimin Huang and Marco Valtorta
{huang6,mgv}@cse.sc.edu
Department of Computer Science and Engineering
University of South Carolina

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, Pearl, Huang, and Valtorta (Tian & Pearl 2002a; 2002b; 2003; Huang & Valtorta 2006a) 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 (Pearl 1995), by providing a sound and complete algorithm for identifiability.

Introduction

This paper focuses on the feasibility of inferring the strength of cause-and-effect relationships from a causal graph (Pearl 1995) (Pearl 2000), 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 causal 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 graphical approaches 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 $P_T(S)$ is identifiable or not. (Pearl and Tian used notation $P(s|do(t))$ and $P(s|\hat{t}\ )$ in (Pearl 2000) and $P_t(s)$ in (Tian & Pearl 2002b), (Tian & Pearl 2003).) Those results are summarized in (Pearl 2000). For example, "back-door" and "front-door" criteria and do-calculus (Pearl 1995); graphical criteria to identify $P_T(S)$ when T is a singleton (Galles & Pearl 1995); 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 (Pearl & Robins 1995).

Recently, Tian and Pearl published a series of papers related to this topic (Tian & Pearl 2002a; 2002b; 2003). 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 (Tian & Pearl 2002b), then to use Algorithm 2 in (Tian & Pearl 2003) (henceforth, the *Identify* algorithm) to calculate the causal effects we want to know. Here, semi-Markovian graphs are defined as causal graphs in which each unobservable variable is a root and has exactly two observable children. Semi-Markovian graphs are sometimes defined differently.

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 (Tian & Pearl 2002b) 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 was no formal proof for this equivalence. Second, the completeness question of the Identify algorithm in (Tian & Pearl 2003) was still open, so that it was unknown whether a causal effect was identifiable if that Identify algorithm failed.

Following Tian and Pearl's work, Huang and Valtorta (2006a) solved the second question. They showed that the Identify algorithm 2 Tian and Pearl used on semi-Markovian models is sound and complete. A similar result was also obtained by Shpitser and Pearl(2006) independently.

In this paper, we focus on general causal graphs directly and our proofs show, as Tian and Pearl pointed out, that Algorithm 2 in (Tian & Pearl 2003) can also be used in general causal models, and we prove that the algorithm is sound and complete.

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 identifiability 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.

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*. A causal Bayesian network in which each hidden variable is a root node with exactly two observed children is called a *semi-Markovian model*.

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:

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

We note explicitly that the post-intervention distribution $P_t(v)$ 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:

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

The post-intervention distribution $P_t(n)$ is defined as:

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

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, $P_t(s)$ is given by:

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

We give out a formal definition of *identifiability* below, which follows (Tian & Pearl 2003).

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 captures the intuition that, given the causal graph G, in an identifiable model, the quantity $P_t(s)$ can be determined from the observed distribution $P(n)$ alone.

Normally, when we use S and T, we think they are both observable variable subsets of N and mutually disjoint. So, s is always consistent with t in 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, $T = N \backslash S$, for convenience and for uniformity with (Tian & Pearl 2002b), we define

$$Q[S] = P_{N \backslash S}(S) \tag{7}$$

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

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, the path $V_1 \leftarrow U_i \rightarrow V_2$ is called a *bidirected link*. 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.

Any causal Bayesian network may be transformed to one in which each unobservable variable is an ancestor of one or more observable variables in such a way that the answer to an identifiability question is preserved. Details of this transformation are given in (Huang & Valtorta 2006a). In this paper, we assume that this transformation has taken place.

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 unobservable parent set* $DUP(C)$ as below. A node V belongs to $DUP(C)$ if and only if both of these two conditions are satisfied: i) V is an unobservable node; ii) there is a directed path from V to an element of C such all the internal nodes on that path are unobservable nodes.

For a given observable variable set $C \subseteq N$, let G_C denote 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., $S = An(S) \cap N$.

Lemmas

In this section we present five lemmas that will be used in the next two sections. The first two lemmas are proved in (Tian & Pearl 2002b). The other three are proved in (Huang & Valtorta 2006b).

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

$$\sum_{V_i \in C \backslash W} Q[C] = Q[W] \tag{8}$$

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

$$Q[H] = \prod_{i=1}^{n} Q[H_i] \tag{9}$$

(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

$$Q[H_i] = \prod_{\{j | V_{h_j} \in H_i\}} \frac{Q[H^{(j)}]}{Q[H^{(j-1)}]} \tag{10}$$

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

$$Q[H^{(j)}] = \sum_{h \backslash h^{(j)}} Q[H] \tag{11}$$

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-component 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 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 obtained by removing a directed or bidirected edge from G.*

Intuitively, this lemma states that unidentifiability does not change by adding any links.

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 nodes in $S \backslash S_1 \cup T \backslash T_1$, then $P_T(S)$ is not identifiable in the graph G.*

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 $Q[B]_{G_B}$.*

Identify Algorithm For $Q[S]$

Based on the lemmas in the last section, we give out an algorithm to calculate $Q[S]$, which is a transfered version of the similar algorithm in (Tian & Pearl 2003). Here $S \subset N$ is a subset of observable variables.

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 G_S, $S_j = S'_j \cap S, 1 \leqslant j \leqslant l$.

Based on lemma 2, for any model on graph G, we have

$$Q[S] = \prod_{j=1}^{l} Q[S_j] \tag{12}$$

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}.*

Proof: Only if part: From lemma 5, each $Q[S_j]$ is identifiable in G_{N_j} means each $Q[S_j]$ is identifiable from $Q[N_j]$ on 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 12, $Q[S]$ is identifiable.

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

We give out the algorithm (which follows (Tian & Pearl 2003)) to get $Q[C]$ form $Q[T]$.

Algorithm 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 $Q[C] = \sum_{T \setminus C} Q[T]$ (Cf. lemma 1)

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 $Q[A] = \sum_{T \setminus A} Q[T]$ (Cf. lemma 2)*
3. *Output Identify(C,T_1,$Q[T_1]$)*

We obtain that the problem of whether $Q[C]$ is computable from $Q[T]$ is reduced to that of whether $Q[C]$ is computable from $Q[T_1]$.

Using lemma 5, we know $Q[C]$ is computable from $Q[T]$ in G_T if and only if $Q[C]$ is identifiable form $Q[T_1]$ in graph G_{T_1}.

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

Proof: We know this lemma is true when the models are semi-Markovian (Huang & Valtorta 2006a) (Shpitser & Pearl 2006). And any general Markovian model with graph G can be transformed to a semi-Markovian model with graph $PJ(G, N)$ through the following a projection (Verma 1993): 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 $X \rightarrow Y$ to $PJ(G, N)$ (if it does not exist yet)
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 $PJ(G, N)$

If model G and $S \in N(G)$ satisfy the conditions of lemma 7, then, $PJ(G, N(G))$ and S satisfy those conditions too. So we just need to prove that if $Q[S]$ is unidentifiable in $PJ(G, N)$ then $Q[S]$ is unidentifiable in G.

$Q[S]$ is unidentifiable in $PJ(G, N)$ means we have two models M_1 and M_2 on graph $PJ(G, N)$ that satisfy $P^{M_1}(n) = P^{M_2}(n) > 0$, but $Q^{M_1}[S] \neq Q^{M_2}[S]$.

Based on M_1 and M_2, we construct two models M'_1 and M'_2 on a subgraph of G. We assume the state space for each node V_i in $PJ(G, N)$ is $S(V_i)$.

We define a state space set $SS(X)$ for each node X in $V(G)$ and set them to be empty at the beginning.

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, then 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}. Select the shortest divergent path between X and Y in G and add the state space of U_{xy} to the state space set of internal nodes on that path if it is not in them yet.

For any observable node X in $PJ(G, N)$, we denote the set of all X's parents' state space as $SPa(X)$. We define the state space of each node in G' as the product of its state space set. Then 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 $\sum_{Y \in Pa(X)} SS(Y)$.

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

$$P^{M'_k}(X = x | SPa(X) = a, (PB(X) - SPa(X)) = b) = P^{M_k}(X = x | SPa(X) = a) \tag{13}$$

If the same node state space in $SPa(X)$ appears more than once on $PB(X)$, then we 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. The CPT of X is defined as

$$P^{M'_k}(y_1, \ldots, y_n, z_1, \ldots, z_m | y'_1, \ldots, y'_n, b) = \begin{cases} \prod_{Z_i \in \{Z_1, \ldots, Z_m\}} P^{M_k}(Z_i = z_i) & all\ y_j = y'_j \\ 0 & exist\ y_j \neq y'_j \end{cases} \quad (14)$$

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 arbitrarily select one of them in the above definition.

Based on this definition, we have $P^{M_k}(n) = P^{M'_k}(n) > 0$ and $Q^{M'_1}[S] \neq Q^{M'_2}[S]$. □

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.*

The two lemmas below follow from theorem 1.

Lemma 8 *If* $S \subset N$ *in graph* G*,* e *is a link exiting one* S *node, and graph* G' *is the same as graph* G *except that 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: Since e is a link exiting an S node, 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. From Algorithm Identify(C,T,Q), Algorithm Computing $Q[S]$, and theorem 1, we know that $Q[S]$ 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. □

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*.*

Proof: When $P_T(S)$ is unidentifiable in graph G', there are two models M_1 and M_2 on G' such that: $P^{M_1}(n) = P^{M_2}(n) > 0$, but for given (s, t), $P_t^{M_1}(s) = a > P_t^{M_2}(s) = b > 0$. Assume in that s, $X_1 = x_1$, $X_2 = x_2$.

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 $P(F(x_1) = 0) = 0.5$; for any $x \in S(X_1)$,$x \neq x_1$, $P(F(x) = 0) = (a - b)/4$. $P(F(x) = 0) + P(F(x) = 1) = 1$ for all x in $S(X_1)$.

For any node X, which is not in $\{U_1, \ldots, U_k, X_2\}$, we define for $i = 1, 2$ the state space for X in model M'_k to be the state space of X in model M_k. For any node X, which is in $\{U_1, \ldots, U_k\}$, 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 $S(X_1)$. The state space of X_2 in M'_k is defined as $S(X_2) \times \{0, 1\}$.

For any node X that is not in $\{U_1, \ldots, U_k, X_2\}$ and has no parent in $\{U_1, \ldots, U_k, X_2\}$, its CPT in M'_k is the same as the CPT in M_k. For any node X, that is not in $\{U_1, \ldots, U_k, X_2\}$ but has some parent in $\{U_1, \ldots, U_k, X_2\}$, then its own state space is the same as in M_k but some of its parents' state spaces are changed. It is simple to insure that this change does not effect the CPT: we omit the details.

For u_1 and $x_1$we define

$$P^{M'_i}((u_1, x_1) | pa(U_1), x'_1) = \begin{cases} P^{M_i}(u_1 | pa(U_1)) & x_1 = x'_1 \\ 0 & x_1 \neq x'_1 \end{cases} \quad (15)$$

For u_i, which is an instance of $U_i \in \{U_2, \ldots, U_k\}$, we define

$$P^{M'_i}((u_i, x_1) | pa'(U_i), (u_{i-1}, x'_1)) = \begin{cases} P^{M_i}(u_i | pa'(U_i), u_{i-1}) & x_1 = x'_1 \\ 0 & x_1 \neq x'_1 \end{cases} \quad (16)$$

For x_2, which is an instance of X_2,$m = 0, 1$, we define

$$P^{M'_i}((x_2, m) | pa'(X_2), (u_k, x_1)) = P^{M_i}(x_2 | pa'(X_2), u_k) \times P(F(x_1) = m) \quad (17)$$

Then for any instance n of N in model M'_1 and M'_2,

$$P^{M'_1}(n) = P^{M'_2}(n) > 0 \quad (18)$$

But for $(s \backslash \{x_2\}, (x_2, 0), t)$,

$$P_t^{M'_1}(s \backslash \{x_1\}) > 0.5a \quad (19)$$

$$P_t^{M'_2}(s \backslash \{x_1\}) < 0.5b + (a - b)/4 < 0.5a \quad (20)$$

From models M'_1 and M'_2, we conclude that $P_T(S \backslash \{X_1\})$ is unidentifiable in G. □

We define the *s-ancestor set* D of S in G to be an observable variable set for which $S \subseteq D \subseteq N$ and $D = An(S)$ in G_D.

Lemma 11 *If* 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: The if part is easy since, if $Q[D]$ is identifiable, $\sum_{D\setminus S} Q[D]$ is identifiable.

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

Because D is an s-ancestor set of S, we can find an order of nodes in $D\setminus S$, say $X_1, \ldots, X_k$, such that 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 from X_i in G that does not exist in G' is e_i. And graph G_i is obtained by adding link e_i to graph G_{i-1}, starting with $G_0 = G'$.

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

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

What we want to compute is:

$$P_t(s) = \sum_{N\setminus(T\cup S)} P_t(n\setminus t) = \sum_{N\setminus(T\cup S)} Q[N\setminus T] \qquad (21)$$

Let $D = An(S)_{G_{N\setminus T}} \cap N$. D is an ancestral set in graph $G_{N\setminus T}$, Lemma 1 allows us to conclude that $\sum_{N\setminus(T\cup D)} Q[N\setminus T] = Q[D]$. Therefore, we have:

$$P_t(s) = \sum_{D\setminus S} \sum_{N\setminus(T\cup D)} Q[N\setminus T] = \sum_{D\setminus S} Q[D] \qquad (22)$$

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

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* $D = An(S)_{G_{N\setminus T}} \cap N$
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* $P_T(S) = \sum_{D\setminus S} Q[D]$

Our discussion above shows,

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

Conclusion

In this paper, we review the identify algorithm for semi-Markovian graphs given by J.Tian and J.Pearl. 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 graphical do calculus by J.Pearl in (Pearl 2000).

References

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

Huang, Y., and Valtorta, M. 2006a. On the completeness of an identifiability algorithm for semi-markovian models, TR-2006-001. Technical report, University of South Carolina Department of Computer Science. available at http://www.cse.sc.edu/ mgv/reports/tr2006-001.pdf.

Huang, Y., and Valtorta, M. 2006b. A study of identifiability in causal Bayesian networks, TR-2006-002. Technical report, University of South Carolina Department of Computer Science. available at http://www.cse.sc.edu/ mgv/reports/tr2006-002.pdf.

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

Pearl, J. 1995. Causal diagrams for empirical research. *Biometrika* 82:669–710.

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

Shpitser, I., and Pearl, J. 2006. Identification of joint interventional distributions in recursive semi-markovian causal models, R-327. Technical report, Cognitive Systems Laboratory, University of California at Los Angeles. available at http://ftp.cs.ucla.edu/pub/stat_ser/r327.pdf.

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

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

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

Verma, T. S. 1993. Graphical aspects of causal models, R-191. Technical report, Cognitive Systems Laboratory, University of California at Los Angeles.