Learning Semi-Markovian Causal Models using Experiments

Stijn Meganck[1], Sam Maes[2], Philippe Leray[2] and Bernard Manderick[1]
[1]CoMo
Vrije Universiteit Brussel
Brussels, Belgium
[2]LITIS
INSA Rouen
St. Etienne du Rouvray, France

Abstract

Semi-Markovian causal models (SMCMs) are an extension of causal Bayesian networks for modeling problems with latent variables. However, there is a big gap between the SMCMs used in theoretical studies and the models that can be learned from observational data alone. The result of standard algorithms for learning from observations, is a complete partially ancestral graph (CPAG), representing the Markov equivalence class of maximal ancestral graphs (MAGs). In MAGs not all edges can be interpreted as immediate causal relationships. In order to apply state-of-the-art causal inference techniques we need to completely orient the learned CPAG and to transform the result into a SMCM by removing non-causal edges. In this paper we combine recent work on MAG structure learning from observational data with causal learning from experiments in order to achieve that goal. More specifically, we provide a set of rules that indicate which experiments are needed in order to transform a CPAG to a completely oriented SMCM and how the results of these experiments have to be processed. We will propose an alternative representation for SMCMs that can easily be parametrised and where the parameters can be learned with classical methods. Finally, we show how this parametrisation can be used to develop methods to efficiently perform both probabilistic and causal inference.

1 Introduction

This paper discusses graphical models that can handle latent variables without explicitly modeling them quantitatively. For such problem domains, several paradigms exist, such as *semi-Markovian causal models* or *maximal ancestral graphs*. Applying these techniques to a problem domain consists of several steps, typically: structure learning from observational and experimental data, parameter learning, probabilistic inference, and, quantitative causal inference.

A problem is that each existing approach only focuses on one or a few of all the steps involved in the process of modeling a problem including latent variables. The goal of this paper is to investigate the integral process from learning from observational and experimental data unto different types of efficient inference.

Semi-Markovian causal models (SMCMs) (Pearl, 2000; Tian and Pearl, 2002) are specifically suited for performing quantitative causal inference in the presence of latent variables. However, at this time no efficient parametrisation of such models is provided and there are no techniques for performing efficient probabilistic inference. Furthermore there are no techniques for learning these models from data issued from observations, experiments or both.

Maximal ancestral graphs (MAGs), developed in (Richardson and Spirtes, 2002) are specifically suited for structure learning from observational data. In MAGs every edge depicts an ancestral relationship. However, the techniques only learn up to Markov equivalence

and provide no clues on which additional experiments to perform in order to obtain the fully oriented causal graph. See (Eberhardt et al., 2005; Meganck et al., 2006) for that type of results on Bayesian networks without latent variables. Furthermore, no parametrisation for discrete variables is provided for MAGs (only one in terms of Gaussian distributions) and no techniques for probabilistic and causal inference have been developed.

We have chosen to use SMCMs in this paper, because they are the only formalism that allows to perform causal inference while taking into account the influence of latent variables. However, we will combine existing techniques to learn MAGs with newly developed methods to provide an integral approach that uses both observational data and experiments in order to learn fully oriented semi-Markovian causal models.

In this paper we also introduce an alternative representation for SMCMs together with a parametrisation for this representation, where the parameters can be learned from data with classical techniques. Finally, we discuss how probabilistic and quantitative causal inference can be performed in these models.

The next section introduces the necessary notations and definitions. It also discusses the semantical and other differences between SMCMs and MAGs. In section 3, we discuss structure learning for SMCMs. Then we introduce a new representation for SMCMs that can easily be parametrised. We also show how both probabilistic and causal inference can be performed with the help of this new representation.

2 Notation and Definitions

We start this section by introducing notations and defining concepts necessary in the rest of this paper. We will also clarify the differences and similarities between the semantics of SMCMs and MAGs.

2.1 Notation

In this work uppercase letters are used to represent variables or sets of variables, i.e., $V = \{V_1, \ldots, V_n\}$, while corresponding lowercase letters are used to represent their instantiations, i.e., v_1, v_2 and v is an instantiation of all v_i. $P(V_i)$ is used to denote the probability distribution over all possible values of variable V_i, while $P(V_i = v_i)$ is used to denote the probability distribution over the instantiation of variable V_i to value v_i. Usually, $P(v_i)$ is used as an abbreviation of $P(V_i = v_i)$.

The operators $Pa(V_i), Anc(V_i), Ne(V_i)$ denote the observable parents, ancestors and neighbors respectively of variable V_i in a graph and $Pa(v_i)$ represents the values of the parents of V_i. Likewise, the operator $LPa(V_i)$ represents the latent parents of variable V_i. If $V_i \leftrightarrow V_j$ appears in a graph then we say that they are spouses, i.e., $V_i \in Sp(V_j)$ and vice versa.

When two variables V_i, V_j are independent we denote it by $(V_i \perp\!\!\!\perp V_j)$, when they are dependent by $(V_i \not\perp\!\!\!\perp V_j)$.

2.2 Semi-Markovian Causal Models

Consider the model in Figure 1(a), it is a problem with observable variables $V_1, \ldots, V_6$ and latent variables L_1, L_2 and it is represented by a directed acyclic graph (DAG). As this DAG represents the actual problem, henceforth we will refer to it as the **underlying DAG**.

The central graphical modeling representation that we use are the semi-Markovian causal models (Tian and Pearl, 2002).

Definition 1. *A **semi-Markovian causal model** (SMCM) is a representation of a causal Bayesian network (CBN) with observable variables $V = \{V_1, \ldots, V_n\}$ and latent variables $L = \{L_1, \ldots, L_{n'}\}$. Furthermore, every latent variable has no parents (i.e., is a root node) and has exactly two children that are both observed.*

See Figure 1(b) for an example SMCM representing the underlying DAG in (a).

In a SMCM each directed edge represents an immediate autonomous causal relation between the corresponding variables. Our operational definition of causality is as follows: a relation from variable C to variable E is causal in a certain context, when a manipulation in the form of a randomised controlled experiment on vari-

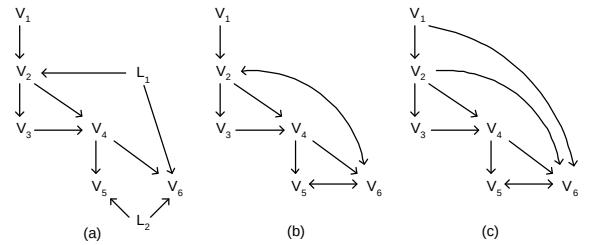

Figure 1: (a) A problem domain represented by a causal DAG model with observable and latent variables. (b) A semi-Markovian causal model representation of (a). (c) A maximal ancestral graph representation of (a).

able C, induces a change in the probability distribution of variable E, in that specific context (Neapolitan, 2003).

In a SMCM a bi-directed edge between two variables represents a latent variable that is a common cause of these two variables. Although this seems very restrictive, it has been shown that models with arbitrary latent variables can be converted into SMCMs while preserving the same independence relations between the observable variables (Tian and Pearl, 2002).

The semantics of both directed and bi-directed edges imply that SMCMs are not maximal. This means that they do not represent all dependencies between variables with an edge. This is because in a SMCM an edge either represents an immediate causal relation or a latent common cause, and therefore dependencies due to a so called inducing path, will not be represented by an edge.

A node is a collider on a path if both its immediate neighbors on the path point into it.

Definition 2. *An **inducing path** is a path in a graph such that each observable non-endpoint node of the path is a collider, and an ancestor of at least one of the endpoints.*

Inducing paths have the property that their endpoints can not be separated by conditioning on any subset of the observable variables. For instance, in Figure 1(a), the path $V_1 \rightarrow V_2 \leftarrow L_1 \rightarrow V_6$ is inducing.

SMCMs are specifically suited for another type of inference, i.e., causal inference.

Definition 3. ***Causal inference** is the process of calculating the effect of manipulating some variables X on the probability distribution of some other variables Y; this is denoted as $P(Y = y|do(X = x))$.*

An example causal inference query in the SMCM of Figure 1(b) is $P(V_6 = v_6|do(V_2 = v_2))$.

Tian and Pearl have introduced theoretical causal inference algorithms to perform causal inference in SMCMs (Pearl, 2000; Tian and Pearl, 2002). However these algorithms assume that any distribution that can be obtained from the JPD over the observable variables is available. We will show that their algorithm performs efficiently on our parametrisation.

2.3 Maximal Ancestral Graphs

Maximal ancestral graphs are another approach to modeling with latent variables (Richardson and Spirtes, 2002). The main research focus in that area lies on learning the structure of these models.

Ancestral graphs (AGs) are complete under marginalisation and conditioning. We will only discuss AGs without conditioning as is commonly done in recent work.

Definition 4. *An **ancestral graph** without conditioning is a graph containing directed $\rightarrow$ and bi-directed $\leftrightarrow$ edges, such that there is no bi-directed edge between two variables that are connected by a directed path.*

Pearl's d-separation criterion can be extended to be applied to ancestral graphs, and is called m-separation in this setting. M-separation characterizes the independence relations represented by an ancestral graph.

Definition 5. *An ancestral graph is said to be **maximal** if, for every pair of non-adjacent nodes (V_i, V_j) there exists a set Z such that V_i and V_j are independent conditional on Z.*

A non-maximal AG can be transformed into a MAG by adding some bi-directed edges (indicating confounding) to the model. See Figure 1(c) for an example MAG representing the same model as the underlying DAG in (a).

In this setting a directed edge represents an ancestral relation in the underlying DAG with latent variables. I.e., an edge from variable A to B represents that in the underlying causal DAG with latent variables, there is a directed path between A and B.

Bi-directed edges represent a latent common cause between the variables. However, if there is a latent common cause between two variables A and B, and there is also a directed path between A and B in the underlying DAG, then in the MAG the ancestral relation takes precedence and a directed edge will be found between the variables. $V_2 \rightarrow V_6$ in Figure 1(c) is an example of such an edge.

Furthermore, as MAGs are maximal, there will also be edges between variables that have no immediate connection in the underlying DAG, but that are connected via an inducing path. The edge $V_1 \rightarrow V_6$ in Figure 1(c) is an example of such an edge.

These semantics of edges make causal inference in MAGs virtually impossible. As stated by the *Manipulation Theorem* (Spirtes et al., 2000), in order to calculate the causal effect of a variable A on another variable B, the immediate parents (i.e., the pre-intervention causes) of A have to be removed from the model. However, as opposed to SMCMs, in MAGs an edge does not necessarily represent an immediate causal relationship, but rather an ancestral relationship and hence in general the modeler does not know which are the real immediate causes of a manipulated variable.

An additional problem for finding the pre-intervention causes of a variable in MAGs is that when there is an ancestral relation and a latent common cause between variables, then the ancestral relation takes precedence and the confounding is absorbed in the ancestral relation.

Complete partial ancestral graphs (CPAGs) are defined in (Zhang and Spirtes, 2005) in the following way.

Definition 6. *Let $[G]$ be the Markov equivalence class for an arbitrary MAG G. The **complete partial ancestral graph** (CPAG) for $[G]$, P_G, is a graph with possibly the following edges $\rightarrow, \leftrightarrow, o\!\!-\!\!o, o\!\!\rightarrow$, such that*

1. *P_G has the same adjacencies as G (and hence any member of $[G]$) does;*
2. *A mark of arrowhead $>$ is in P_G if and only if it is invariant in $[G]$; and*
3. *A mark of tail $-$ is in P_G if and only if it is invariant in $[G]$.*
4. *A mark of o is in P_G if not all members in $[G]$ have the same mark.*

2.4 Assumptions

As is customary in the graphical modeling research area, the SMCMs we take into account in this paper are subject to some simplifying assumptions:

1. *Stability*, i.e., the independencies in the CBN with observed and latent variables that generates the data are structural and not due to several influences exactly canceling each other out (Pearl, 2000).

2. Only a *single immediate connection* per two variables in the underlying DAG. I.e., we do not take into account problems where two variables that are connected by an immediate causal edge are also confounded by a latent variable causing both variables. Constraint based learning techniques such as IC* (Pearl, 2000) and FCI (Spirtes et al., 2000) also do not explicitly recognise multiple edges between variables. However, in (Tian and Pearl, 2002) Tian presents an algorithm for performing causal inference where such relations between variables are taken into account.

3. *No selection bias.* Mimicking recent work, we do not take into account latent variables that are conditioned upon, as can be the consequence of selection effects.

4. *Discrete variables.* All the variables in our models are discrete.

3 Structure learning

Just as learning a graphical model in general, learning a SMCM consists of two parts: structure learning and parameter learning. Both can be done using data, expert knowledge and/or experiments. In this section we discuss only structure learning.

3.1 Without latent variables

Learning the structure of Bayesian networks without latent variables has been studied by a number of researchers (Pearl, 2000; Spirtes et al., 2000). The results of applying one of those algorithms is a representative of the Markov equivalence class.

In order to perform probabilistic or causal inference, we need a fully oriented structure. For probabilistic inference this can be any representative of the Markov equivalence class, but for causal inference we need the correct causal graph that models the underlying system. In order to achieve this, additional experiments have to be performed.

In previous work (Meganck et al., 2006), we studied learning the completely oriented structure for causal Bayesian networks without latent variables. We proposed a solution to minimise the total cost of the experiments needed by using elements from decision theory. The techniques used there could be extended to the results of this paper.

3.2 With latent variables

In order to learn graphical models with latent variables from observational data, the Fast Causal Inference (FCI) algorithm (Spirtes et al., 2000) has been proposed. Recently this result has been extended with the complete tail augmentation rules introduced in (Zhang and Spirtes, 2005). The results of this algorithm is a CPAG, representing the Markov equivalence class of MAGs consistent with the data.

As mentioned above for MAGs, in a CPAG the directed edges have to be interpreted as being ancestral instead of causal. This means that there is a directed edge from V_i to V_j if V_i is an ancestor of V_j in the underlying DAG

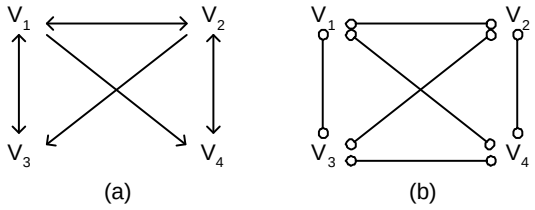

Figure 2: (a) A SMCM. (b) Result of FCI, with an i-false edge $V_3 o\!\!-\!\!o V_4$.

and there is no subset of observable variables D such that $(V_i \perp\!\!\!\perp V_j | D)$. This does not necessarily mean that V_i has an immediate causal influence on V_j: it may also be a result of an inducing path between V_i and V_j. For instance in Figure 1(c), the link between V_1 and V_6 is present due to the inducing path V_1, V_2, L_1, V_6 shown in Figure 1(a).

Inducing paths may also introduce an edge of type $o\!\!\rightarrow$ or $o\!\!-\!\!o$ between two variables, indicating either a directed or bi-directed edge, although there is no immediate influence in the form of an immediate causal influence or latent common cause between the two variables. An example of such a link is $V_3 o\!\!-\!\!o V_4$ in Figure 2.

A consequence of these properties of MAGs and CPAGs is that they are not suited for causal inference, since the immediate causal parents of each observable variable are not available, as is necessary according to the Manipulation Theorem. As we want to learn models that can perform causal inference, we will discuss how to transform a CPAG into a SMCM in the next sections. Before we start, we have to mention that we assume that the CPAG is correctly learned from data with the FCI algorithm and the extended tail augmentation rules, i.e., each result that is found is not due to a sampling error.

3.3 Transforming the CPAG

Our goal is to transform a given CPAG in order to obtain a SMCM that corresponds to the correspondig DAG. Remember that in general there are three types of edges in a CPAG: $\rightarrow$, $o\!\!\rightarrow$, $o\!\!-\!\!o$, in which o means either a tail mark $-$ or a directed mark $>$. So one of the tasks to

obtain a valid SMCM is to disambiguate those edges with at least one o as an endpoint. A second task will be to identify and remove the edges that are created due to an inducing path.

In the next section we will first discuss exactly which information we obtain from performing an experiment. Then, we will discuss the two possibilities $o\!\!\rightarrow$ and $o\!\!-\!\!o$. Finally, we will discuss how we can find edges that are created due to inducing paths and how to remove these to obtain the correct SMCM.

3.3.1 Performing experiments

The experiments discussed here play the role of the manipulations that define a causal relation (cf. Section 2.2). An experiment on a variable V_i, i.e., a randomised controlled experiment, removes the influence of other variables in the system on V_i. The experiment forces a distribution on V_i, and thereby changes the joint distribution of all variables in the system that depend directly or indirectly on V_i but does not change the conditional distribution of other variables given values of V_i. After the randomisation, the associations of the remaining variables with V_i provide information about which variables are influenced by V_i (Neapolitan, 2003). When we perform the experiment we cut all influence of other variables on V_i. Graphically this corresponds to removing all incoming edges into V_i from the underlying DAG.

All parameters besides those for the variable experimented on, (i.e., $P(V_i|Pa(V_i))$), remain the same. We then measure the influence of the manipulation on variables of interest to get the post-interventional distribution on these variables.

To analyse the results of the experiment we compare for each variable of interest V_j the original distribution P and the post-interventional distribution P_E, thus comparing $P(V_j)$ and $P_E(V_j) = P(V_j|do(V_i = v_i))$.

We denote performing an experiment at variable V_i or a set of variables W by $exp(V_i)$ or $exp(W)$ respectively, and if we have to condition on some other set of variables D while performing the experiment, we denote it as $exp(V_i)|D$ and $exp(W)|D$.

In general if a variable V_i is experimented on and another variable V_j is affected by this experiment, we say that V_j *varies with* $exp(V_i)$, denoted by $exp(V_i) \rightsquigarrow V_j$. If there is no variation in V_j we note $exp(V_i) \not\rightsquigarrow V_j$.

Although conditioning on a set of variables D might cause some variables to become probabilistically dependent, conditioning will not influence whether two variables vary with each other when performing an experiment. I.e., suppose the following structure is given $V_i \rightarrow D \leftarrow V_j$, then conditioning on D will make V_i dependent on V_j, but when we perform an experiment on V_i and check whether V_j varies with V_i then conditioning on D will make no difference.

First we have to introduce the following definition:

Definition 7. *A **potentially directed path** (p.d. path) in a CPAG is a path made only of edges of types $o\!\!\rightarrow$ and $\rightarrow$, with all arrowheads in the same direction. A p.d. path from V_i to V_j is denoted as $V_i \dashrightarrow V_j$.*

3.3.2 Solving $o\!\!\rightarrow$

An overview of the different rules for solving $o\!\!\rightarrow$ is given in Table 1

Type 1(a) Given	 $Ao\!\!\rightarrow B$ $exp(A) \not\rightsquigarrow B$
Action	$A \leftrightarrow B$
Type 1(b) Given	 $Ao\!\!\rightarrow B$ $exp(A) \rightsquigarrow B$ $\nexists$p.d. path $(length \geq 2) A \dashrightarrow B$
Action	$A \rightarrow B$
Type 1(c) Given	 $Ao\!\!\rightarrow B$ $exp(A) \rightsquigarrow B$ $\exists$p.d. path $(length \geq 2) A \dashrightarrow B$
Action	Block all p.d. paths by conditioning on blocking set D (a) $exp(A)\|D \rightsquigarrow B$: $A \rightarrow B$ (b) $exp(A)\|D \not\rightsquigarrow B$: $A \leftrightarrow B$

Table 1: An overview of the different actions needed to complete edges of type $o\!\!\rightarrow$.

For any edge $V_i o\!\!\rightarrow V_j$, there is no need to perform an experiment on V_j because we know that there can be no immediate influence of V_j on V_i, so we will only perform an experiment on V_i.

If $exp(V_i) \not\rightsquigarrow V_j$, then there is no influence of V_i on V_j, so we know that there can be no directed edge between V_i and V_j and thus the only remaining possibility is $V_i \leftrightarrow V_j$ (Type 1(a)).

If $exp(V_i) \rightsquigarrow V_j$, then we know for sure that there is an influence of V_i on V_j, we now need to discover whether this influence is immediate or via some intermediate variables. Therefore we make a difference whether there is a potentially directed (p.d.) path between V_i and V_j of length ≥ 2, or not. If no such path exists, then the influence has to be immediate and the edge is found $V_i \rightarrow V_j$ (Type 1(b)).

If at least one p.d. path $V_i \dashrightarrow V_j$ exists, we need to block the influence of those paths on V_j while performing the experiment, so we try to find a blocking set D for all these paths. If $exp(V_i)|D \rightsquigarrow V_j$, then the influence has to be immediate, because all paths of length ≥ 2 are blocked, so $V_i \rightarrow V_j$. On the other hand if $exp(V_i)|D \not\rightsquigarrow V_j$, there is no immediate influence and the edge is $V_i \leftrightarrow V_j$ (Type 1(c)).

3.3.3 Solving *o–o*

An overview of the different rules for solving *o–o* is given in Table 2.

For any edge $V_i o\!\!-\!\!o V_j$, we have no information at all, so we might need to perform experiments on both variables.

If $exp(V_i) \not\rightsquigarrow V_j$, then there is no influence of V_i on V_j so we know that there can be no directed edge between V_i and V_j and thus the edge is of the following form: $V_i \leftarrow\!\!o V_j$, which then becomes a problem of Type 1.

If $exp(V_i) \rightsquigarrow V_j$, then we know for sure that there is an influence of V_i on V_j, and as in the case of Type 1(b), we make a difference whether there is a potentially directed path between V_i and V_j of length ≥ 2, or not. If no such path exists, then the influence has to be immediate and the edge must be turned into $V_i \rightarrow V_j$.

If at least one p.d. path $V_i \dashrightarrow V_j$ exists, we need to block the influence of those paths on V_j while performing the experiment, so we find a blocking set D like with Type 1(c). If $exp(V_i)|D \rightsquigarrow V_j$, then the influence has to be immediate, because all paths of length ≥ 2 are blocked, so $V_i \rightarrow V_j$. On the other hand if $exp(V_i)|D \not\rightsquigarrow V_j$, there is no immediate influence and the edge is of type: $V_i \leftarrow\!\!o V_j$, which again becomes a problem of Type 1.

Type 2(a) Given	 $Ao\!\!-\!\!oB$ $exp(A) \not\rightsquigarrow B$
Action	$A \leftarrow\!\!oB$(⇒Type 1)
Type 2(b) Given	 $Ao\!\!-\!\!oB$ $exp(A) \rightsquigarrow B$ ∄p.d. path $(length \geq 2) A \dashrightarrow B$
Action	$A \rightarrow B$
Type 2(c) Given	 $Ao\!\!-\!\!oB$ $exp(A) \rightsquigarrow B$ ∃p.d. path $(length \geq 2) A \dashrightarrow B$
Action	Block all p.d. paths by conditioning on blocking set D (a) $exp(A)\|D \rightsquigarrow B$: $A \rightarrow B$ (b) $exp(A)\|D \not\rightsquigarrow B$: $A \leftarrow\!\!oB$ (⇒Type 1)

Table 2: An overview of the different actions needed to complete edges of type *o–o*.

3.3.4 Removing inducing path edges

An inducing path between two variables V_i and V_j might create an edge between these two variables during learning because the two are dependent conditional on any subset of observable variables. As mentioned before, this type of edges is not present in SMCMs as it does not represent an immediate causal influence or a latent variable in the underlying DAG. We will call such an edge an *i-false* edge.

For instance, in Figure 1(a) the path V_1, V_2, L_1, V_6 is an inducing path, which causes the FCI algorithm to find an i-false edge between V_1 and V_6, see Figure 1(c). Another example is given in Figure 2 where the SMCM is given in (a) and the result of FCI in (b). The

edge between V_3 and V_4 in (b) is a consequence of the inducing path via the observable variables V_3, V_1, V_2, V_4.

In order to be able to apply a causal inference algorithm we need to remove all i-false edges from the learned structure. We need to identify the substructures that can indicate this type of edges. This is easily done by looking at any two variables that are connected by an immediate connection, and when this edge is removed, they have at least one inducing path between them. To check whether the immediate connection needs to be present we have to block all inducing paths by performing one or more experiments on an inducing path blocking set (i-blocking set) D^{ip} and block all other paths by conditioning on a blocking set D. If V_i and V_j are dependent, i.e., $V_i \not\perp\!\!\!\perp V_j$ under these circumstances the edge is correct and otherwise it can be removed.

In the example of Figure 1(c), we can block the inducing path by performing an experiment on V_2, and hence can check that V_1 and V_6 do not covary with each other in these circumstances, so the edge can be removed.

In Table 3 an overview of the actions to resolve i-false edges is given.

Given	A pair of connected variables V_i, V_j A set of inducing paths $V_i, \ldots, V_j$
Action	Block all inducing paths $V_i, \ldots, V_j$ by conditioning on i-blocking set D^{ip}. Block all other paths between V_i and V_j by conditioning on blocking set D. When performing all $exp(D^{ip})\|D$: if $V_i \not\perp\!\!\!\perp V_j$: confounding is real else remove edge between V_i, V_j

Table 3: Removing inducing path edges.

3.4 Example

We will demonstrate a number of steps to discover the completely oriented SMCM (Figure 1(b)) based on the result of the FCI algorithm applied on observational data generated from the underlying DAG in Figure 1(a). The result

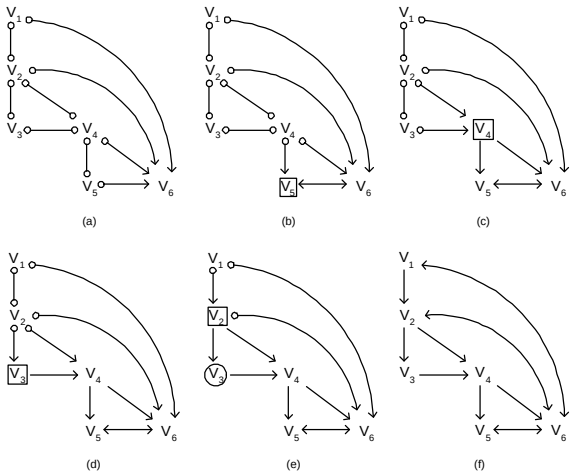

Figure 3: (a) The result of FCI on data of the underlying DAG of Figure 1(a). (b) Result of an experiment on V_5. (c) After experiment on V_4. (d) After experiment on V_3. (e) After experiment on V_2 while conditioning on V_3. (f) After resolving all problems of Type 1 and 2.

of the FCI algorithm can be seen in Figure 3(a). We will first resolve problems of Type 1 and 2, and then remove i-false edges. The result of each step is indicated in Figure 3.

- $exp(V_5)$
  - $V_5 o\!\!-\!\!o V_4$:
    $exp(V_5) \not\rightarrow V_4 \Rightarrow V_4 o\!\!\rightarrow V_5$ (Type 2(a))
  - $V_5 o\!\!\rightarrow V_6$:
    $exp(V_5) \not\rightarrow V_6 \Rightarrow V_5 \leftrightarrow V_6$ (Type 1(a))
- $exp(V_4)$
  - $V_4 o\!\!-\!\!o V_2$:
    $exp(V_4) \not\rightarrow V_2 \Rightarrow V_2 o\!\!\rightarrow V_4$ (Type 2(a))
  - $V_4 o\!\!-\!\!o V_3$:
    $exp(V_4) \not\rightarrow V_3 \Rightarrow V_3 o\!\!\rightarrow V_4$ (Type 2(a))
  - $V_4 o\!\!\rightarrow V_5$:
    $exp(V_4) \rightsquigarrow V_5 \Rightarrow V_4 \rightarrow V_5$ (Type 1(b))
  - $V_4 o\!\!\rightarrow V_6$:
    $exp(V_4) \rightsquigarrow V_6 \Rightarrow V_4 \rightarrow V_6$ (Type 1(b))
- $exp(V_3)$
  - $V_3 o\!\!-\!\!o V_2$:
    $exp(V_3) \not\rightarrow V_2 \Rightarrow V_2 o\!\!\rightarrow V_3$ (Type 2(a))
  - $V_3 o\!\!\rightarrow V_4$:
    $exp(V_3) \rightsquigarrow V_4 \Rightarrow V_3 \rightarrow V_4$ (Type 1(b))

- $exp(V_2)$ and $exp(V_2)|V_3$, because two p.d. paths between V_2 and V_4
  - $V_2 o\!\!-\!\!o V_1$: $exp(V_2) \not\rightsquigarrow V_1 \Rightarrow V_1 o\!\!\rightarrow V_2$ (Type 2(a))
  - $V_2 o\!\!\rightarrow V_3$: $exp(V_2) \rightsquigarrow V_3 \Rightarrow V_2 \rightarrow V_3$ (Type 1(b))
  - $V_2 o\!\!\rightarrow V_4$: $exp(V_2)|V_3 \rightsquigarrow V_4 \Rightarrow V_2 \rightarrow V_4$ (Type 1(c))

After resolving all problems of Type 1 and 2 we end up with the SMCM structure shown in Figure 3(f). This representation is no longer consistent with the MAG representation since there are bi-directed edges between two variables on a directed path, i.e., V_2, V_6. There is a potentially i-false edge $V_1 \leftrightarrow V_6$ in the structure with inducing path V_1, V_2, V_6, so we need to perform an experiment on V_2, blocking all other paths between V_1 and V_6 (this is also done by $exp(V_2)$ in this case). Given that the original structure is as in Figure 1(a), performing $exp(V_2)$ shows that V_1 and V_6 are independent, i.e., $exp(V_2) : (V_1 \perp\!\!\!\perp V_6)$. Thus the bi-directed edge between V_1 and V_6 is removed, giving us the SMCM of Figure 1(b).

4 Parametrisation of SMCMs

As mentioned before, in their work on causal inference, Tian and Pearl provide an algorithm for performing causal inference given knowledge of the structure of an SMCM and the joint probability distribution (JPD) over the observable variables. However, they do not provide a parametrisation to efficiently store the JPD over the observables.

We start this section by discussing the factorisation for SMCMs introduced in (Tian and Pearl, 2002). From that result we derive an additional representation for SMCMs and a parametrisation that facilitates probabilistic and causal inference. We will also discuss how these parameters can be learned from data.

4.1 Factorising with Latent Variables

Consider an underlying DAG with observable variables $V = \{V_1, \ldots, V_n\}$ and latent variables $L = \{L_1, \ldots, L_{n'}\}$. Then the joint probability distribution can be written as the following mixture of products:

$$P(v) = \sum_{\{l_k | L_k \in L\}} \prod_{V_i \in V} P(v_i | Pa(v_i), LPa(v_i)) \prod_{L_j \in L} P(l_j). \quad (1)$$

Taking into account that in a SMCM the latent variables are implicitly represented by bi-directed edges, we introduce the following definition.

Definition 8. *In a SMCM, the set of observable variables can be partitioned by assigning two variables to the same group iff they are connected by a bi-directed path. We call such a group a* ***c-component*** *(from "confounded component") (Tian and Pearl, 2002).*

For instance, in Figure 1(b) variables V_2, V_5, V_6 belong to the same c-component. Then it can be readily seen that c-components and their associated latent variables form respective partitions of the observable and latent variables. Let $Q[S_i]$ denote the contribution of a c-component with observable variables $S_i \subset V$ to the mixture of products in Equation 1. Then we can rewrite the JPD as follows: $P(v) = \prod_{i \in \{1,\ldots,k\}} Q[S_i]$.

Finally, (Tian and Pearl, 2002) proved that each $Q[S]$ could be calculated as follows. Let $V_{o_1} < \ldots < V_{o_n}$ be a topological order over V, and let $V^{(i)} = V_{o_1} < \ldots < V_{o_i}$, $i = 1, \ldots, n$, and $V^{(0)} = \emptyset$.

$$Q[S] = \prod_{V_i \in S} P(v_i | (T_i \cup Pa(T_i)) \backslash \{V_i\}) \quad (2)$$

where T_i is the c-component of the SMCM G reduced to variables $V^{(i)}$, that contains V_i. The SMCM G reduced to a set of variables $V' \subset V$ is the graph obtained by removing from the graph all variables $V \backslash V'$ and the edges that are connected to them.

In the rest of this section we will develop a method for deriving a DAG from a SMCM.

We will show that the classical factorisation $\prod P(v_i|Pa(v_i))$ associated with this DAG, is the same as the one associated with the SMCM, as above.

4.2 Parametrised representation

Here we first introduce an additional representation for SMCMs, then we show how it can be parametrised and, finally we discuss how this new representation could be optimised.

4.2.1 PR-representation

Consider $V_{o_1} < \ldots < V_{o_n}$ to be a topological order O over the observable variables V, only considering the directed edges of the SMCM, and let $V^{(i)} = V_{o_1} < \ldots < V_{o_i}$, $i = 1, \ldots, n$, and $V^{(0)} = \emptyset$. Then Table 4 shows how the parametrised (PR-) representation can be obtained from the original SMCM structure.

Given a SMCM G and a topological ordering O, the PR-representation has these properties:
1. The nodes are V, i.e., the observable variables of the SMCM.
2. The directed edges are the same as in the SMCM.
3. The confounded edges are replaced by a number of directed edges as follows: Add an edge from node V_i to node V_j iff: a) $V_i \in (T_j \cup Pa(T_j))$, where T_j is the c-component of G reduced to variables $V^{(j)}$ that contains V_j, b) and there was not already an edge between nodes V_i and V_j.

Table 4: Obtaining the PR-representation.

This way each variable becomes a child of the variables it would condition on in the calculation of the contribution of its c-component as in Equation (2).

Figure 4(a) shows the PR-representation of the SMCM in Figure 1(a). The topological order that was used here is $V_1 < V_2 < V_3 < V_4 < V_5 < V_6$ and the directed edges that have been added are $V_1 \rightarrow V_5$, $V_2 \rightarrow V_5$, $V_1 \rightarrow V_6$, $V_2 \rightarrow V_6$, and, $V_5 \rightarrow V_6$.

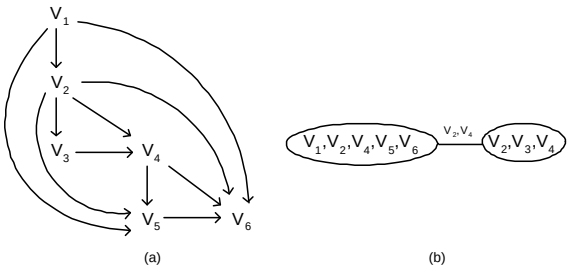

Figure 4: (a) The PR-representation applied to the SMCM of Figure 1(b). (b) Junction tree representation of the DAG in (a).

The resulting DAG is an I-map (Pearl, 1988), over the observable variables of the independence model represented by the SMCM. This means that all the independencies that can be derived from the new graph must also be present in the JPD over the observable variables. This property can be more formally stated as the following theorem.

Theorem 9. *The PR-representation PR derived from a SMCM S is an **I-map** of that SMCM.*

Proof. Consider two variables A and B in PR that are not connected by an edge. This means that they are not independent, since a necessary condition for two variables to be conditionally independent in a stable DAG is that they are not connected by an edge. PR is stable as we consider only stable problems (Assumption 1 in Section 2.4).

Then, from the method for constructing PR we can conclude that S (i) contains no directed edge between A and B, (ii) contains no bi-directed edge between A and B, (iii) contains no inducing path between A and B. Property (iii) holds because the only inducing paths that are possible in a SMCM are those between a member of a c-component and the immediate parent of another variable of the c-component, and in these cases the method for constructing PR adds an edge between those variables. Because of (i),(ii), and (iii) we can conclude that A and B are independent in S. □

4.2.2 Parametrisation

For this DAG we can use the same parametrisation as for classical BNs, i.e., learning $P(v_i|Pa(v_i))$ for each variable, where $Pa(v_i)$ denotes the parents in the new DAG. In this way the JPD over the observable variables factorises as in a classical BN, i.e., $P(v) = \prod P(v_i|Pa(v_i))$. This follows immediately from the definition of a c-component and from Equation (2).

4.2.3 Optimising the Parametrisation

We have mentioned that the number of edges added during the creation of the PR-representation depends on the topological order of the SMCM.

As this order is not unique, choosing an order where variables with a lesser amount of parents have precedence, will cause less edges to be added to the DAG. This is because most of the added edges go from parents of c-component members to c-component members that are topological descendants.

By choosing an optimal topological order, we can conserve more conditional independence relations of the SMCM and thus make the graph more sparse, thus leading to a more efficient parametrisation.

4.2.4 Learning parameters

As the PR-representation of SMCMs is a DAG as in the classical Bayesian network formalism, the parameters that have to be learned are $P(v_i|Pa(v_i))$. Therefore, techniques such as ML and MAP estimation (Heckerman, 1995) can be applied to perform this task.

4.3 Probabilistic inference

One of the most famous existing probabilistic inference algorithm for models without latent variables is the *junction tree* algorithm (JT) (Lauritzen and Spiegelhalter, 1988).

These techniques cannot immediately be applied to SMCMs for two reasons. First of all until now no efficient parametrisation for this type of models was available, and secondly, it is not clear how to handle the bi-directed edges that are present in SMCMs.

We have solved this problem by first transforming the SMCM into its PR-representation, which allows us to apply the junction tree inference algorithm. This is a consequence of the fact that, as previously mentioned, the PR-representation is an I-map over the observable variables. And as the JT algorithm is based only on independencies in the DAG, applying it to an I-map of the problem gives correct results. See Figure 4(b) for the junction tree obtained from the parametrised representation in Figure 4(a).

Note that any other classical probabilistic inference technique that only uses conditional independencies between variables could also be applied to the PR-representation.

4.4 Causal inference

Tian and Pearl (2002) developed an algorithm for performing causal inference, however as mentioned before they have not provided an efficient parametrisation.

Richardson and Spirtes (2003) show causal inference in AGs on an example, but a detailed approach is not provided and the problem of what to do when some of the parents of a variable are latent is not solved.

By definition in the PR-representation, the parents of each variable are exactly those variables that have to be conditioned on in order to obtain the factor of that variable in the calculation of the c-component, see Table 4 and (Tian and Pearl, 2002). Thus, the PR-representation provides all the necessary quantitative information, while the original structure of the SMCM provides the necessary structural information, for the algorithm by Tian and Pearl to be applied.

5 Conclusions and Perspectives

In this paper we have proposed a number of solutions to problems that arise when using SMCMs in practice.

More precisely, we showed that there is a big gap between the models that can be learned from data alone and the models that are used in theory. We showed that it is important to retrieve the fully oriented structure of a SMCM,

and discussed how to obtain this from a given CPAG by performing experiments.

For future work we would like to relax the assumptions made in this paper. First of all we want to study the implications of allowing two types of edges between two variables, i.e., confounding as well as a immediate causal relationship. Another direction for possible future work would be to study the effect of allowing multiple joint experiments in other cases than when removing inducing path edges.

Furthermore, we believe that applying the orientation and tail augmentation rules of (Zhang and Spirtes, 2005) after each experiment, might help to reduce the number of experiments needed to fully orient the structure. In this way we could extend to SMCMs our previous results (Meganck et al., 2006) on minimising the total number of experiments in causal models without latent variables. This would allow to compare empirical results with the theoretical bounds developed in (Eberhardt et al., 2005).

Up until now SMCMs have not been parametrised in another way than by the entire joint probability distribution. Another contribution of our paper is that we showed that using an alternative representation, we can parametrise SMCMs in order to perform probabilistic as well as causal inference. Furthermore this new representation allows to learn the parameters using classical methods.

We have informally pointed out that the choice of a topological order when creating the PR-representation, influences its size and thus its efficiency. We would like to investigate this property in a more formal manner. Finally, we have started implementing the techniques introduced in this paper into the structure learning package (SLP)[1] of the Bayesian networks toolbox (BNT)[2] for MATLAB.

Acknowledgments

This work was partially funded by an IWT-scholarship and by the IST Programme of the European Community, under the PASCAL network of Excellence, IST-2002-506778.

[1] http://banquiseasi.insa-rouen.fr/projects/bnt-slp/

[2] http://bnt.sourceforge.net/

References

F. Eberhardt, C. Glymour, and R. Scheines. 2005. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In *Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 178–183. Edinburgh.

D. Heckerman. 1995. A tutorial on learning with Bayesian networks. Technical report, Microsoft Research.

S. L. Lauritzen and D. J. Spiegelhalter. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, series B*, 50:157–244.

S. Meganck, P. Leray, and B. Manderick. 2006. Learning causal Bayesian networks from observations and experiments: A decision theoretic approach. In *Modeling Decisions in Artificial Intelligence, LNAI 3885*, pages 58 – 69.

R.E. Neapolitan. 2003. *Learning Bayesian Networks*. Prentice Hall.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

J. Pearl. 2000. *Causality: Models, Reasoning and Inference*. MIT Press.

T. Richardson and P. Spirtes. 2002. Ancestral graph Markov models. Technical Report 375, Dept. of Statistics, University of Washington.

T. Richardson and P. Spirtes, 2003. *Causal inference via ancestral graph models*, chapter 3. Oxford Statistical Science Series: Highly Structured Stochastic Systems. Oxford University Press.

P. Spirtes, C. Glymour, and R. Scheines. 2000. *Causation, Prediction and Search*. MIT Press.

J. Tian and J. Pearl. 2002. On the identification of causal effects. Technical Report (R-290-L), UCLA C.S. Lab.

J. Zhang and P. Spirtes. 2005. A characterization of Markov equivalence classes for ancestral graphical models. Technical Report 168, Dept. of Philosophy, Carnegie-Mellon University.