Causal graphical models with latent variables: learning and inference

Philippe Leray, Stijn Meganck, Sam Maes, Bernard Manderick

Causal Graphical Models with Latent Variables: Learning and Inference

Philippe Leray[1], Stijn Meganck[2], Sam Maes[3], and Bernard Manderick[2]

[1] LINA Computer Science Lab UMR6241, Knowledge and Decision Team, Université de Nantes, France philippe.leray@univ-nantes.fr
[2] Computational Modeling Lab, Vrije Universiteit Brussel, Belgium
[3] LITIS Computer Science, Information Processing and Systems Lab EA4108, INSA Rouen, France

1 Introduction

This chapter discusses causal graphical models for discrete variables that can handle latent variables without explicitly modeling them quantitatively. In the *uncertainty in artificial intelligence* area there exist several paradigms for such problem domains. Two of them are *semi-Markovian causal models* and *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.

We will start this chapter by introducing causal graphical models without latent variables and then move on to models with latent variables.

We will discuss the problem that each of the existing approaches for causal modeling with latent variables only focuses on one or a few of all the steps involved in a generic knowledge discovery approach. The goal of this chapter is to investigate the integral process from observational and experimental data unto different types of efficient inference.

Semi-Markovian causal models (SMCMs) are an approach developed by (Pearl, 2000; Tian and Pearl, 2002a). They 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 to learn these models from data issued from observations, experiments or both.

Maximal ancestral graphs (MAGs) are an approach developed by (Richardson and Spirtes, 2002). They are specifically suited for structure learning in the presence of latent variables from observational data. 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 for Bayesian networks without latent variables. Furthermore, as of yet no parametrisation for discrete variables is provided for MAGs and no techniques for probabilistic inference have been developed. There is some work on algorithms for causal inference, but it is restricted to causal inference quantities that are the same for an entire Markov equivalence class of MAGs (Spirtes et al., 2000; Zhang, 2006).

We have chosen to use SMCMs as a final representation in our work, because they are the only formalism that allows to perform causal inference while fully 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.

Furthermore, we have developed an alternative representation for the probability distribution represented by a SMCM, 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 with the help of the alternative representation and its associated parametrisation[4].

The next section introduces the simplest causal models and their importance. Then we discuss causal models with latent variables. In section 4, we discuss structure learning for those models and in the next section we introduce techniques for learning a SMCM with the help of experiments. Then we propose 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 Importance of Causal Models

We start this section by introducing basic notations necessary for the understanding of the rest of this chapter. Then we will discuss classical probabilistic Bayesian networks followed by causal Bayesian networks. Finally we handle the difference between probabilistic and causal inference, or observation vs. manipulation.

2.1 Notations

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

[4] By the term parametrisation we understand the definition of a complete set of parameters that describes the joint probability distribution which can be efficiently used in computer implementations of probabilistic inference, causal inference and learning algorithms.

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 of 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. 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 Probabilistic Bayesian Networks

Here we briefly discuss classical probabilistic Bayesian networks.

See Figure 1 for a famous example adopted from Pearl (1988) representing an alarm system. The alarm can be triggered either by a burglary, by an earthquake, or by both. The alarm going of might cause John and/or Mary to call the house owner at his office.

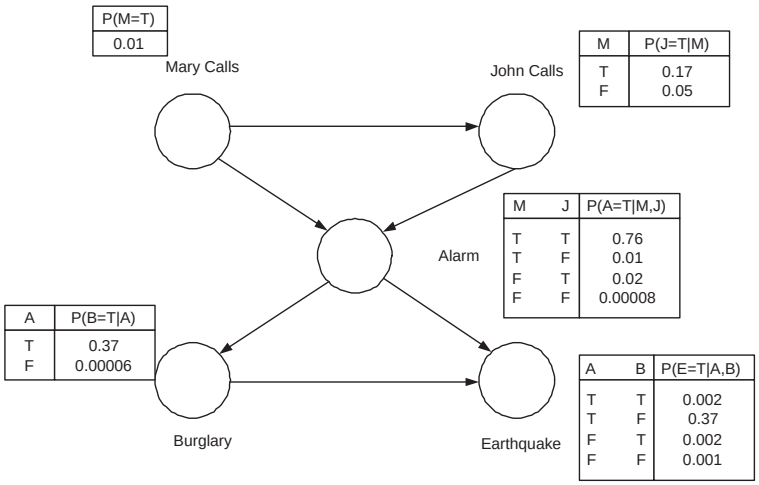

Fig. 1. Example of a Bayesian network representing an alarm system.

In Pearl (1988); Russell and Norvig (1995) probabilistic Bayesian networks are defined as follows:

Definition 1. *A **Bayesian network** is a triple $\langle V, G, P(v_i|Pa(v_i))\rangle$, with:*

- $V = \{V_1, \ldots, V_n\}$*, a set of observable discrete random variables*

- *a directed acyclic graph (DAG) G, where each node represents a variable from V*
- *parameters: conditional probability distributions (CPD) $P(v_i|Pa(v_i))$ of each variable V_i from V conditional on its parents in the graph G.*

The CPDs of a BN represent a factorization of the joint probability distribution as a product of conditional probability distributions of each variable given its parents in the graph:

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

Inference

A BN also allows to efficiently answer probabilistic queries such as

$$P(burglary = true|Johncalls = true, Marycalls = false),$$

in the alarm example of Figure 1. It is the probability that there was a burglary, given that we know John called and Mary did not.

Methods have been developed for efficient exact probabilistic inference when the networks are sparse (Pearl, 1988). For networks that are more complex this is not tractable, and approximate inference algorithms have been formulated (Jordan, 1998), such as variational methods (Jordan et al., 1999) and Monte Carlo methods (Mackay, 1999).

Structure Learning

There are two main approaches for learning the structure of a BN from data: *score-based* learning (Heckerman, 1995) and *constraint-based* learning (Spirtes et al., 2000; Pearl, 2000).

For score-based learning, the goal is to find the graph that best matches the data by introducing a scoring function that evaluates each network with respect to the data, and then to search for the best network according to this score.

Constraint-based methods are based on matching the conditional independence relations observed between variables in the data with those entailed by a graph.

However, in general a particular set of data can be represented by more than one BN. Therefore the above techniques have in common that they can only learn upto the *Markov equivalence class*. Such a class contains all the DAGs that correctly represent the data and for performing probabilistic inference any DAG of the class can be chosen.

2.3 Causal Bayesian Networks

Now we will introduce a category of Bayesian networks where the edges have a causal meaning.

We have previously seen that in general there is more than one probabilistic BN that can be used to represent the same JPD. More specifically, all the members of a given Markov equivalence class can be used to represent the same JPD.

Opposed to that, in the case of a causal Bayesian network (CBN) we assume that in reality there is a single underlying causal Bayesian network that *generates* the JPD. In Figure 2 we see a conceptual sketch: the box represents the real world where a causal Bayesian network generates the data in the form of a joint probability distribution. Below we see the BNs that represent all the independence relations present in the JPD. Only one of them is the causal Bayesian network, in this case the rightmost.

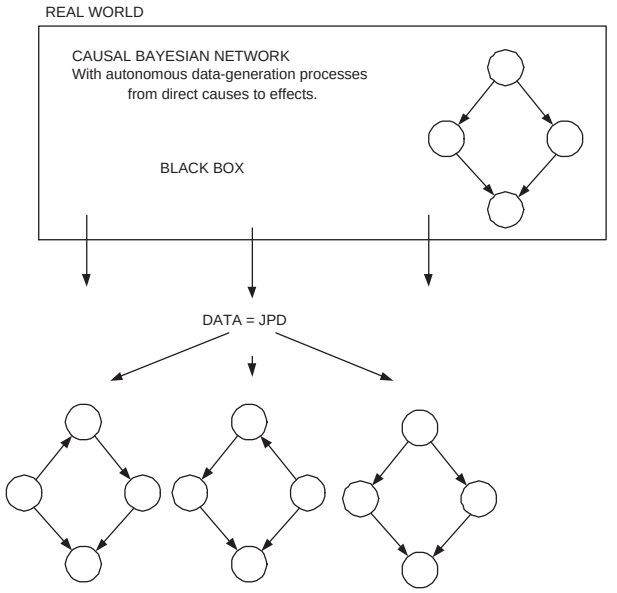

Fig. 2. Conceptual sketch of how a CBN generates a JPD, that in its turn can be represented by several probabilistic BNs of which one is a CBN.

The definition of causal Bayesian networks is as follows:

Definition 2. *A **causal Bayesian network** is a triple* $\langle V, G, P(v_i|Pa(v_i))\rangle$*, with:*

- $V = \{V_1, \ldots, V_n\}$*, a set of observable discrete random variables*

- *a directed acyclic graph (DAG) G, where each node represents a variable from V*
- *parameters: conditional probability distributions (CPD) $P(v_i|Pa(v_i))$ of each variable V_i from V conditional on its parents in the graph G.*
- *Furthermore, the directed edges in G represent an autonomous causal relation between the corresponding variables.*

We see that it is exactly the same as Definition 1 for probabilistic Bayesian networks, with the extra addition of the last item.

This is different from a classical BN, where the arrows only represent a probabilistic dependency, and not necessarily a causal one.

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

This means that in a CBN, each CPD $P(v_i|Pa(v_i))$ represents a stochastic assignment process by which the values of V_i are chosen in response to the values of $Pa(V_i)$ in the underlying domain. This is an approximation of how events are physically related with their effects in the domain that is being modeled. For such an assignment process to be autonomous means that it must stay invariant under variations in the processes governing other variables Pearl (2000).

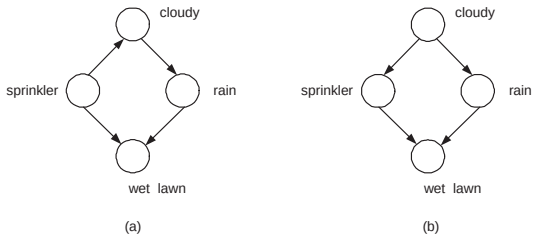

Fig. 3. (a) A BN where not all the edges have a causal meaning. (b) A CBN that can represent the same JPD as (a).

In the BN of Figure 3(a), these assumptions clearly do not hold for all edges and nodes, since in the underlying physical domain, whether or not it is cloudy is not caused by the state of the variable *sprinkler*, i.e. whether or not the sprinkler is on.

Moreover, one could want to manipulate the system, for example by changing the way in which the state of the sprinkler is determined by its causes. More specifically, by changing how the sprinkler reacts to the cloudiness. In order to incorporate the effect of such a manipulation of the system into the

model, some of the CPDs have to be changed. However, in a non-causal BN, it is not immediately clear which CPDs have to be changed and exactly how this must be done.

In contrast, in Figure 3(b), we see a causal BN that can represent the same JPD as the BN in (a). Here the extra assumptions do hold. For example, if in the system the state of the sprinkler is caused by the cloudiness, and thus the CPD $P(sprinkler|cloudy)$ represents an assignment process that is an approximation of how the sprinkler is physically related to the cloudiness. Moreover, if the sensitivity of the sprinkler is changed, this will only imply a change in the CPD $P(sprinkler|cloudy)$, but not in the processes governing other variables such as $P(rain|cloudy)$.

Note that CBNs are a subclass of BNs and therefore they allow probabilistic inference. In the next section we will discuss what additional type of inference can be performed with them, but first we treat how CBNs can be learned.

Structure Learning

As CBNs are a subset of all BNs, the same techniques as for learning the structure of BNs can be used to learn upto the Markov equivalence class. As mentioned before, for BNs any member of the equivalence can be used.

For CBNs this is not the case, as we look for the orientation of the unique network that can both represent the JPD and the underlying causal influences between the variables. In general, in order to obtain the causal orientation of all the edges, experiments have to be performed, where some variables in the domain are experimentally manipulated and the potential effects on other variables are observed.

Eberhardt et al. (2005) discuss theoretical bounds on the amount of experiments that have to be performed to obtain the full oriented CBN. Meganck et al. (2006) have proposed a solution to learning CBNs from experiments and observations, where the total cost of the experiments is minimised by using elements from decision theory.

Other related approaches include Cooper and Yoo (1999) who derived a Bayesian method for learning from an arbitrary mixture of observational and experimental data.

Tong and Koller (2001) provide an algorithm that actively chooses the experiments to perform based on the model learned so far. In this setting they assume there are a number of query variables Q that can be experimented on and then measure the influence on all other variables $V \backslash Q$. In order to choose the optimal experiment they introduce a loss-function, based on the uncertainty of the direction of an edge, to help indicate which experiment gives the most information. Using the results of their experiments they update the distribution over the possible networks and network parameters. Murphy (2001) introduces a slightly different algorithm of the same approach.

2.4 Causal Inference

Here we will briefly introduce causal inference, we start by pointing out the difference with probabilistic inference, and then move on to discuss an important theorem related to causal inference.

Observation vs. Manipulation

An important issue in reasoning under uncertainty is to distinguish between different types of conditioning, each of which modify a given probability distribution in response to information obtained.

Definition 3. ***Conditioning by observation*** *refers to the way in which a probability distribution of Y should be modified when a modeler passively observes the information $X = x$.*

This is represented by conditional probabilities that are defined as follows:

$$P(Y = y|X = x) = P(y|x) = \frac{P(Y = y, X = x)}{P(X = x)}. \tag{2}$$

This type of conditioning is referred to as *probabilistic inference*. It is used when the modeler wants to predict the behavior of some variables that have not been observed, based on the state of some other variables. E.g. will the patients' infection cause him to have a fever ?

This can be very useful in a lot of situations, but in some cases the modeler does not merely want to predict the future behavior of some variables, but has to decide which action to perform, i.e. which variable to manipulate in which way. For example, will administering a dose of 10mg of antibiotics cure the patients' infection ?

In that case probabilistic inference is not the right technique to use, because in general it will return the level of association between the variables instead of the causal influence. In the antibiotics example: if observing the administration of a dose of 10mg of antibiotics returns a high probability of curing the infection, this can be due to (a mix of) several reasons:

- the causal influence of antibiotics on curing the infection,
- the causal influence of curing the infection on antibiotics,
- the causal influence of another variable on both antibiotics and curing the infection, or,
- the causal influence of both antibiotics and curing the infection on another variable that we inadvertently condition on (i.e. selection bias).

Without extra information we cannot make the difference between these reasons. On the other hand if we want to know whether administering a dose of 10mg of antibiotics will cure the patients' infection, we will need to isolate the causal influence of antibiotics on curing the infection and this process is denoted by *causal inference*.

Definition 4. ***Causal inference*** *is the process of calculating the effect of manipulating some variables X on the probability distribution of some other variables Y.*

Definition 5. ***Conditioning by intervention*** *or* ***manipulation***[5] *refers to the way the distribution Y should be modified if we intervene externally and force the value of X to be equal to x.*

To make the distinction clear, Pearl has introduced the ***do-operator*** *(Pearl, 2000)*[6]*:*

$$P(Y = y|do(X = x)) \tag{3}$$

The manipulations we are treating here are surgical in the sense that they only directly change the variable of interest (X in the case of $X = do(x)$).

To reiterate, it is important to realize that conditioning by observation is typically not the way the distribution of Y should be modified if we intervene externally and force the value of X to be equal to x, as can be seen next:

$$P(Y = y|do(X = x)) \neq P(Y = y|X = x) \tag{4}$$

and the quantity on the left-hand side cannot be calculated from the joint probability distribution $P(v)$ alone, without additional assumptions imposed on the graph, i.e. that a directed edge represents an autonomous causal relation as in CBNs.

Consider the simple CBNs of Figure 4 in the left graph

$$P(y|do(x)) = P(y|x)$$

as X is the only immediate cause of Y, but

$$P(x|do(y)) = P(x) \neq P(x|y)$$

as there is no direct or indirect causal relation going from Y to X. The equalities above are reversed in the graph to the right, i.e. there it holds that $P(y|do(x)) = P(y) \neq P(y|x)$ and $P(x|do(y)) = P(x|y)$.

Fig. 4. Two simple causal Bayesian networks.

Next we introduce a theorem that specifies how a manipulation modifies the JPD associated with a CBN.

[5] Throughout this chapter the terms *intervention* and *manipulation* are used interchangeably.

[6] In the literature other notations such as $P(Y = y||X = x)$, $P_{X=x}(Y = y)$, or $P(Y = y|X = \hat{x})$ are abundant.

Manipulation Theorem

Performing a manipulation in a domain that is modeled by a CBN, does modify that domain and the JPD that is used to model it. Before introducing a theorem that specifies how a CBN and the JPD that is associated with it must be changed to incorporate the change induced by a manipulation, we will offer an intuitive example.

Example 1. Imagine we want to disable the alarm in the system represented by the CBN of Figure 5(a) by performing the manipulation *do(alarm=off)*.

This CBN represents an alarm system against burglars, it can be triggered by a burglary, an earthquake or both. Furthermore, the alarm going off might cause the neighbors to call the owner at his work.

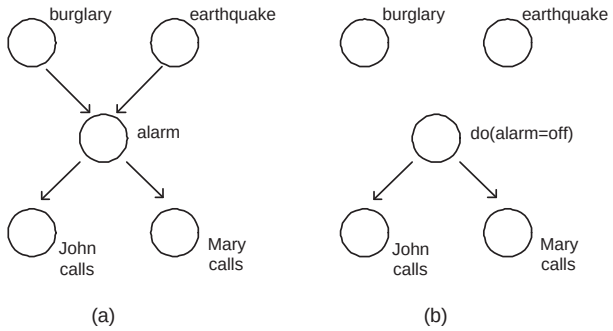

Fig. 5. (a) A CBN of an alarm system. (b) The CBN of the alarm system of (a) after disabling the alarm via an external manipulation: *do(alarm=off)*.

Such a manipulation changes the way in which the value of alarm is being produced in the real world. Originally, the value of alarm was being decided by its immediate causes in the model of Figure 5(a): *burglary* and *earthquake*.

After manipulating the alarm by disabling it, *burglary* and *earthquake* are no longer the causes of the alarm, but have been replaced by the manipulation.

In Figure 5(b) the graph of the post-manipulation CBN is shown. There we can see that the links between the original causes of *alarm* have been severed and that the value of *alarm* has been instantiated to *off*.

To obtain the post-manipulation distribution after fixing a set of variables $M \subseteq V$ to fixed values $M = m$, the factors with the variables in M conditional on their parents in the graph (i.e. their causes in the pre-intervention distribution), have to be removed from the JPD. Formally these are : $P(m_i|Pa(m_i))$ for all variables $M_i \in M$. This is because after the intervention, it is this intervention rather than the parent variables in the graph that cause the values

of the variables in M. Furthermore the remaining occurrences of M in the JPD have to be instantiated to $M = m$.

A manipulation of this type only has a local influence in the sense that only the incoming links of a manipulated variable have to be removed from the model, no factors representing other links have to be modified, except for instantiating the occurrences of the manipulated variables M to m. This is a consequence of the assumption of CBNs that the factors of the JPD represent assignment processes that must stay invariant under variations in the processes governing other variables. Formally, we get from (Spirtes et al., 2000):

Theorem 1. *Given a CBN with variables $V = V_1, \ldots, V_n$ and we perform the manipulation $M = m$ for a subset of variables $M \subseteq V$, the post-manipulation distribution becomes:*

$$P(v|do(m)) = \prod_{V_i \in V \setminus M} P(v_i|Pa(v_i))\Big|_{M=m} \tag{5}$$

Where $|_{M=m}$ stands for instantiating all the occurrences of the variables M to values m in the equation that precedes it.

3 Causal Models with Latent Variables

In all the above we made the assumption of *causal sufficiency*, i.e. that for every variable of the domain that is a common cause, observational data can be obtained in order to learn the structure of the graph and the CPDs. Often this assumption is not realistic, as it is not uncommon that a subset of all the variables in the domain is never observed. We refer to such a variable as a *latent* variable.

We start this section by briefly discussing different approaches to modeling latent variables. After that we introduce two specific models for modeling latent variables and the causal influences between the observed variables. These will be the two main formalisms used in the rest of this chapter so we will discuss their semantics and specifically their differences in a lot of detail.

3.1 Modeling Latent Variables

Consider the model in Figure 6(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**.

One way to represent such a problem is by using this DAG representation and modeling the latent variables explicitly. Quantities for the observable

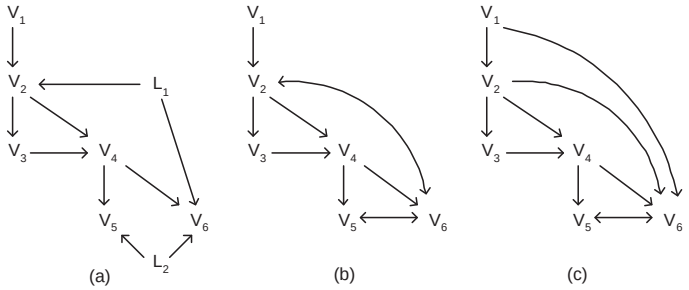

Fig. 6. (a) A problem domain represented by a causal DAG model with observable and latent variables. (b) A semi-Markovian causal model representation of (a). (c) A maximal ancestral graph representation of (a).

variables can then be obtained from the data in the usual way. Quantities involving latent variables however will have to be estimated. This involves estimating the cardinality of the latent variables and this whole process can be difficult and lengthy. One of the techniques to learn models in such a way is the structural EM algorithm (Friedman, 1997).

Another method to take into account latent variables in a model is by representing them implicitly. With that approach, no values have to be estimated for the latent variables, instead their influence is absorbed in the distributions of the observable variables. In this methodology, we only keep track of the position of the latent variable in the graph if it would be modeled, without estimating values for it. Both the modeling techniques that we will use in this chapter belong to that approach, they will be described in the next two sections.

3.2 Semi-Markovian Causal Models

The central graphical modeling representation that we use are the semi-Markovian causal models. They were first used by Pearl (2000), and Tian and Pearl (2002a) have developed causal inference algorithms for them.

Definitions

Definition 6. *A **semi-Markovian causal model** (SMCM) is an acyclic causal graph G with both directed and bi-directed edges. The nodes in the graph represent observable variables $V = \{V_1, \ldots, V_n\}$ and the bi-directed edges implicitly represent latent variables $L = \{L_1, \ldots, L_{n'}\}$.*

See Figure 6(b) for an example SMCM representing the underlying DAG in (a).

The fact that a bi-directed edge represents a latent variable, implies that the only latent variables that can be modeled by a SMCM can not have any parents (i.e. is a root node) and has exactly two children that are both observed. This seems very restrictive, however 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, 2002b).

Semantics

In a SMCM, each directed edge represents an immediate autonomous causal relation between the corresponding variables, just as was the case for causal Bayesian networks.

In a SMCM, a bi-directed edge between two variables represents a latent variable that is a common cause of these two variables.

The semantics of both directed and bi-directed edges imply that SMCMs are not maximal, meaning that not all dependencies between variables are represented by an edge between the corresponding variables. 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.

Definition 7. *An **inducing path** is a path in a graph such that each observable non-endpoint node 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 6(a), the path $V_1 \rightarrow V_2 \leftarrow L_1 \rightarrow V_6$ is inducing.

Parametrisation

SMCMs cannot be parametrised in the same way as classical Bayesian networks (i.e. by the set of CPTs $P(V_i|Pa(V_i))$), since variables that are connected via a bi-directed edge have a latent variable as a parent.

For example in Figure 6(b), choosing $P(V_5|V_4)$ as a parameter to be associated with variable V_5 would only lead to erroneous results, as the dependence with variable V_6 via the latent variable L_2 in the underlying DAG is ignored. As mentioned before, using $P(V_5|V_4, L_2)$ as a parametrisation and estimating the cardinality and the values for latent variable L_2 would be a possible solution. However we choose not to do this as we want to leave the latent variables implicit for reasons of efficiency.

In (Tian and Pearl, 2002a), a factorisation of the joint probability distribution over the observable variables of an SMCM was introduced. Later in this chapter we will derive a representation for the probability distribution represented by a SMCM based on that result.

Learning

In the literature no algorithm for learning the structure of an SMCM exists, in this chapter we introduce techniques to perform that task, given some simplifying assumptions, and with the help of experiments.

Probabilistic Inference

Since as of yet no efficient parametrisation for SMCMs is provided in the literature, no algorithm for performing probabilistic inference exists. We will show how existing probabilistic inference algorithms for Bayesian networks can be used together with our parametrisation to perform that task.

Causal Inference

SMCMs are specifically suited for another type of inference, i.e. causal inference. An example causal inference query in the SMCM of Figure 6(a) is $P(V_6 = v_6|do(V_2 = v_2))$.

As seen before, causal inference queries are calculated via the Manipulation Theorem, which specifies how to change a joint probability distribution (JPD) over observable variables in order to obtain the post-manipulation JPD. Informally, it says that when a variable X is manipulated to a fixed value x, the parents of variables X have to be removed by dividing the JPD by $P(X|Pa(X))$, and by instantiating the remaining occurrences of X to the value x.

When all the parents of a manipulated variable are observable, this can always be done. However, in a SMCM some of the parents of a manipulated variable can be latent and then the Manipulation Theorem cannot be directly used to calculate causal inference queries. Some of these causal quantities can be calculated in other ways but some cannot be calculated at all, because the SMCM does not contain enough information.

When a causal query can be unambiguously calculated from a SMCM, we say that it is *identifiable*. More formally:

Definition 8. *The causal effect of variable X on a variable Y is **identifiable** from a SMCM with graph G if $P_{M_1}(y|do(x)) = P_{M_2}(y|do(x))$ for every pair of SMCMs M_1 and M_2 with $P_{M_1}(v) = P_{M_2}(v) > 0$ and $G_{M_1} = G_{M_2}$, where P_{M_i} and G_{M_i} respectively denote the probability distribution and graph associated with the SMCM M_i.*

In Pearl (2000), Pearl describes the *do-calculus*, a set of inference rules and an algorithm that can be used to perform causal inference. More specifically, the goal of do-calculus is to transform a mathematical expression including manipulated variables related to a SMCM into an equivalent expression involving only standard probabilities of observed quantities. Recent work has

shown that do-calculus is complete (Huang and Valtorta, 2006; Shpitser and Pearl, 2006).

Tian and Pearl have introduced theoretical causal inference algorithms to perform causal inference in SMCMs (Pearl, 2000; Tian and Pearl, 2002a). However, these algorithms assume the availability of a subset of all the conditional distributions that can be obtained from the JPD over the observable variables. We will show that with our representation these conditional distributions can be obtained in an efficient way in order to apply this algorithm.

3.3 Maximal Ancestral Graphs

Maximal ancestral graphs are another approach to modeling with latent variables developed by Richardson and Spirtes (2002). The main research focus in that area lies on learning the structure of these models and on representing exactly all the independences between the observable variables of the underlying DAG.

Definitions

Ancestral graphs (AGs) are graphs that are complete under marginalisation and conditioning. We will only discuss AGs without conditioning as is commonly done in recent work (Zhang and Spirtes, 2005b; Tian, 2005; Ali et al., 2005).

Definition 9. *An **ancestral graph** without conditioning is a graph with no directed cycle 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.*

Definition 10. *An ancestral graph is said to be a **maximal ancestral graph** 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 d-separated given Z.*

A non-maximal AG can be transformed into a unique MAG by adding some bi-directed edges (indicating confounding) to the model. See Figure 6(c) for an example MAG representing the same model as the underlying DAG in (a).

Semantics

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 6(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 6(c) is an example of such an edge.

These semantics of edges make some causal inferences in MAGs impossible. As we have discussed before the Manipulation Theorem states that in order to calculate the causal effect of a variable A on another variable B, the immediate parents (i.e. the old 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 original causes of a variable in MAGs is that when there is an ancestral relation and a latent common cause between variables, that the ancestral relation takes precedence and that the confounding is absorbed in the ancestral relation.

Learning

There is a lot of recent research on learning the structure of MAGs from observational data. The Fast Causal Inference (FCI) algorithm (Spirtes et al., 1999), is a constraint based learning algorithm. Together with the rules discussed in Zhang and Spirtes (2005a), the result is a representation of the Markov equivalence class of MAGs. This representative is referred to as a *complete partial ancestral graph* (CPAG) and in Zhang and Spirtes (2005a) it is defined as follows:

Definition 11. *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.*

In the next section we will discuss learning the structure in somewhat more detail.

Parametrisation and Inference

At this time no parametrisation for MAGs with discrete variables exists that represents all the properties of a joint probability distribution, (Richardson and Spirtes, 2002), neither are there algorithms fo probabilistic inference.

As mentioned above, due to the semantics of the edges in MAGs, not all causal inferences can be performed. However, there is an algorithm due to Spirtes et al. (2000) and refined by Zhang (2006), for performing causal inference in some restricted cases. More specifically, they consider a causal effect to be identifiable if it can be calculated from all the MAGs in the Markov equivalence class that is represented by the CPAG and that quantity is equal for all those MAGs. This severely restricts the causal inferences that can be made, especially if more than conditional independence relations are taken into account during the learning process, as is the case when experiments can be performed. In the context of this causal inference algorithm, Spirtes et al. (2000) also discuss how to derive a DAG that is a minimal I-map of the probability distribution represented by a MAG.

In this chapter we introduce a similar procedure, but for a single SMCM instead of for an entire equivalence class of MAGs. In that way a larger class of causal inferences can be calculated, as the quantities do not have to be equal in all the models of the equivalence class.

4 Structure Learning with Latent Variables

Just as learning a graphical model in general, learning a model with latent variables 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 structure learning and we differentiate between learning from observational and experimental data.

4.1 From Observational Data

In order to learn graphical models with latent variables from observational data a constraint based learning algorithm has been developed by Spirtes et al. (1999). It is called the Fast Causal Inference (FCI) algorithm and it uses conditional independence relations found between observable variables to learn a structure.

Recently this result has been extended with the complete tail augmentation rules introduced in Zhang and Spirtes (2005a). The results of this algorithm is a CPAG, representing the Markov equivalence class of MAGs consistent with the data.

Recent work in the area consists of characterising the equivalence class of CPAGs and finding single-edge operators to create equivalent MAGs (Ali and Richardson, 2002; Zhang and Spirtes, 2005a,b). One of the goals of these

advances is to create methods that search in the space of Markov equivalent models (CPAGs) instead of the space of all models (MAGs), mimicking results in the case without latent variables (Chickering, 2002).

As mentioned before for MAGs, in a CPAG the directed edges have to be interpreted as representing ancestral relations instead of immediate causal relations. More precisely, 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 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 6(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 6(a).

Inducing paths may also introduce $\leftrightarrow$, $\rightarrow$, $o\!\!\rightarrow$ or $o\!\!-\!\!o$ between two variables, 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 7.

A consequence of these properties of MAGs and CPAGs is that they are not very suited for general 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 next.

4.2 From Experimental Data

As mentioned above, the result of current state-of-the-art techniques that learn models with implicit latent variables from observational data is a CPAG. This is a representative of the Markov equivalence class of MAGs. Any MAG in that class will be able to represent the same JPD over the observable variables, but not all those MAGs will have all edges with a correct causal orientation.

Furthermore as mentioned in the above, in MAGs the directed edges do not necessarily have an immediate causal meaning as in CBNs or SMCMs, instead they have an ancestral meaning. If it is your goal to perform causal inference, you will need to know the immediate parents to be able to reason about all causal queries. However, edges that are completely oriented but that do not have a causal meaning will not occur in the CPAG, there they will always be of the types $o\!\!\rightarrow$ or $o\!\!-\!\!o$, so orienting them in correct causal way way suffices.

Finally, MAGs are maximal, thus every missing edge must represent a conditional independence. In the case that there is an inducing path between two variables and no edge in the underlying DAG, the result of the current learning algorithms will be to add an edge between the variables. Again, although these type of edges give the only correct representation of the conditional independence relations in the domain, they do not represent an immediate causal relation (if the inducing edge is directed) or a real latent common cause (if the inducing edge is bi-directed). Because of this they could interfere with

causal inference algorithms, therefore we would like to identify and remove these type of edges.

To recapitulate, the goal of techniques aiming at transforming a CPAG must be twofold:

- finding the correct causal orientation of edges that are not completely specified by the CPAG ($o\!\!\rightarrow$ or $o\!\!-\!\!o$), and,
- removing edges due to inducing paths.

In the next section we discuss how these goals can be obtained by performing experiments.

5 From CPAG to SMCM

Our goal is to transform a given CPAG in order to obtain a SMCM that corresponds to the underlying DAG. Remember that in general there are four types of edges in a CPAG: $\leftrightarrow$, $\rightarrow$, $o\!\!\rightarrow$, $o\!\!-\!\!o$, in which o means either a tail mark $-$ or a directed mark $>$. As mentioned before, 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 introduced some simplfying assumptions we have to use in our work. Then we will discuss exactly which information is obtained from performing an experiment. After that, we will discuss the two possible incomplete edges: $o\!\!\rightarrow$ and $o\!\!-\!\!o$. Finally, we will discuss how we can find edges that are created due to inducing paths and how to remove them to obtain the correct SMCM.

5.1 Assumptions

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

1. *Stability*, i.e. the independencies in the underlying CBN with observed and latent variables that generates the data are structural and not due to several influences exactly cancelling 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, Tian and Pearl (2002a) 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.
5. *Correctness.* 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 or insufficient sample size.

5.2 Performing Experiments

The experiments discussed here play the role of the manipulations discussed in Section 2.3 that define a causal relation. 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 V_i influences (Neapolitan, 2003). To perform the actual experiment we have to cut all influence of other variables on V_i. Graphically this corresponds to removing all incoming arrows into V_i from the underlying DAG.

We then measure the influence of the manipulation on variables of interest by obtaining samples from their post-experimental distributions.

More precisely, to analyse the results of an experiment on a variable V_{exp}, we compare for each variable of interest V_j the original observational sample data D_{obs} with the post-experimental sample data D_{exp}. The experiment consists of manipulating the variable V_{exp} to each of its values v_{exp} a sufficient amount of times in order to obtain sample data sets that are large enough to analyse in a statistically sound way. The result of an experiment will be a data set of samples for the variables of interest for each value i of variable $V_{exp} = i$, we will denote such a data set by $D_{exp,i}$.

In order to see whether an experiment on V_{exp} made an influence on another variable V_j, we compare each post-experimental data set $D_{exp,i}$ with the original observational data set D_{obs} (with a statistical test like χ^2). Only if at least one of the data sets is statistically significantly different, we can conclude that variable V_{exp} causally influences variable V_j.

However, this influence does not necessarily have to be immediate between the variables V_{exp} and V_j, but can be mediated by other variables, such as in the underlying DAG: $V_{exp} \rightarrow V_{med} \rightarrow V_j$.

In order to make the difference between a direct influence and a potentially mediated influence via V_{med}, we will no longer compare the complete data sets $D_{exp,i}$ and D_{obs}. Instead, we will divide both data sets in subsets based on the values of V_{med}, or in other words condition on variable V_{med}. Then we compare each of the smaller data sets $D_{exp,i}|v_{med}$ and $D_{obs}|v_{med}$ with each other and this for all values of V_{med}. By conditioning on a potentially

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

Table 1. An overview of how to complete edges of type $o\!\!\rightarrow$.

mediating variable, we block the causal influence that might go through that variable and we obtain the immediate relation between V_{exp} and V_j.

Note that it might seem that if the mediating variable is a collider, this approach will fail, because conditioning on a collider on a path between two variables creates a dependence between those two variables. However, this approach will still be valid and this is best understood with an example: imagine the underlying DAG is of the form $V_{exp} \cdots \rightarrow V_{med} \leftarrow \ldots V_j$. In this case, when we compare each $D_{exp,i}$ and D_{obs} conditional on V_{med}, we will find no significant difference between both data sets, and this for all the values of V_{med}. This is because the dependence that is created between V_{exp} and V_j by conditioning on the collider V_{med} is present in both the original underlying DAG and in the post-experimental DAG, and thus this is also reflected in the data sets $D_{exp,i}$ and D_{obs}.

In order not to overload that what follows with unnecessary complicated notation we will 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 Z on the data obtained by performing the experiment, we denote it as $exp(V_i)|Z$ and $exp(W)|Z$.

In general if a variable V_i is experimented on and another variable V_j is affected by this experiment, i.e. has another distribution after the experiment than before, 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 $exp(V_i) \not\rightsquigarrow V_j$.

Before going to the actual solutions we have to introduce the notion of potentially directed paths:

Definition 12. *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 $V_i \dashrightarrow V_j$.*

5.3 Solving $o\!\!\rightarrow$

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

Ao–oB	Type 2(a)	Type 2(b)	Type 2(c)
Exper. result	$exp(A) \not\rightsquigarrow B$	$exp(A) \rightsquigarrow B$ $\nexists$p.d. path $A \dashrightarrow B$ ($length \geq 2$)	$exp(A) \rightsquigarrow B$ $\exists$p.d. path $A \dashrightarrow B$ ($length \geq 2$)
Orient. result	$A \leftarrow\!\!o B$ ($\Rightarrow$Type 1)	$A \rightarrow B$	Block all p.d. paths by conditioning on blocking set Z: $exp(A)\|Z \rightsquigarrow B$: $A \rightarrow B$ $exp(A)\|Z \not\rightsquigarrow B$: $A \leftarrow\!\!o B$ ($\Rightarrow$Type 1)

Table 2. An overview of how to complete edges of type o–o.

For any edge $V_i o\!\!\rightarrow V_j$, there is no need to perform an experiment at 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 Z for all these paths. If $exp(V_i)|Z \rightsquigarrow V_j$, 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 $exp(V_i)|Z \not\rightsquigarrow V_j$, there is no immediate influence and the edge is $V_i \leftrightarrow V_j$ (Type 1(c)).

A blocking set Z consists of one variable for each p.d. path. This variable can be chosen arbitrarily as we have explained before that conditioning on a collider does not invalidate our experimental approach.

5.4 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 like with 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 becomes $V_i \to V_j$.

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 Z like with Type 1(c). If $exp(V_i)|Z \rightsquigarrow V_j$, then the influence has to be immediate, because all paths of length ≥ 2 are blocked, so $V_i \to V_j$. On the other hand if $exp(V_i)|Z \not\rightsquigarrow V_j$, there is no immediate influence and the edge is of the following form: $V_i \leftarrow\!\!o V_j$, which again becomes a problem of Type 1.

5.5 Removing Inducing Path Edges

In the previous phase only o-parts of edges of a CPAG have been oriented. The graph that is obtained in this way can contain both directed and bi-directed edges, each of which can be of two types. For the directed edges:

- an immediate causal edge that is also present in the underlying DAG
- an edge that is due to an inducing path in the underlying DAG.

For the bi-directed edges:

- an edge that represents a latent variable in the underlying DAG
- an edge that is due to an inducing path in the underlying DAG.

When representing the same underlying DAG, a SMCM and the graph obtained after orienting all unknown endpoints of the CPAG have the same connections except for edges due to inducing paths in the underlying DAG, these edges are only represented in the experimentally oriented graph.

Definition 13. *We will call an edge between two variables V_i and V_j* ***i-false*** *if it was created due to an inducing path, i.e. because the two variables are dependent conditional on any subset of observable variables.*

For instance in Figure 6(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 6(c). Another example is given in Figure 7 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 through 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. The substructures that can indicate this type of edges can be identified by looking at any two variables that a) are connected by an edge, and, b) 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) Z^{ip} and block all other open paths by conditioning on a blocking set Z. Note that the set of variables Z^{ip} are the set of variables which get an assigned value during the experiments, the set

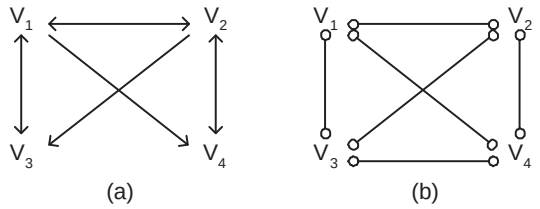

Fig. 7. (a) A SMCM. (b) Result of FCI, with an i-false edge $V_3 o\!\!-\!\!o V_4$.

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

Table 3. Removing i-false edges.

of variables Z are used when looking for independences in the interventional data. If V_i and V_j are dependent, i.e. $(V_i \not\perp\!\!\!\perp V_j)$, under these circumstances then the edge is correct and otherwise it can be removed.

In the example of Figure 6(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.

An i-blocking set consists of a collider on each of the inducing paths connecting the two variables of interest. Here a blocking set Z is a set of variables that blocks each of the other open paths between the two variables of interest.

Table 3 gives an overview of the actions to resolve i-false edges.

5.6 Example

We will demonstrate a number of steps to discover the completely oriented SMCM (Figure 6(b)) based on the result of the FCI algorithm applied on observational data generated from the underlying DAG in Figure 6(a). The result of the FCI algorithm can be seen in Figure 8(a). We will first resolve problems of Type 1 and 2, and then remove i-false edges. The result of each step is explained in Table 4 and indicated in Figure 8.

After resolving all problems of Type 1 and 2 we end up with the structure shown in Figure 8(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. However, this structure is not necessarily a SMCM yet, as there is a potentially i-false edge $V_1 \leftrightarrow V_6$ in the structure

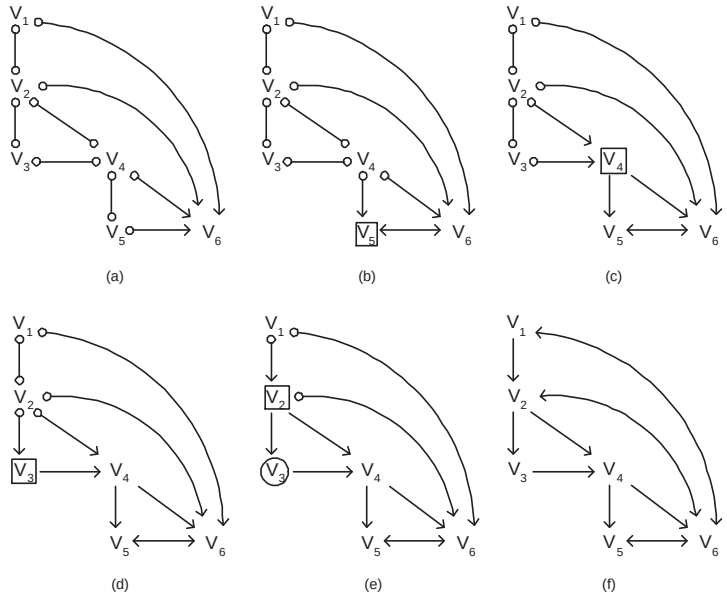

Fig. 8. (a) The result of FCI on data of the underlying DAG of Figure 6(a). (b) Result of an experiment at V_5. (c) Result after experiment at V_4. (d) Result after experiment at V_3. (e) Result after experiment at V_2 while conditioning on V_3. (f) Result of resolving all problems of Type 1 and 2.

Exper.	Edge before	Experiment result	Edge after	Type
$exp(V_5)$	$V_5 o\!\!-\!\!o V_4$	$exp(V_5) \not\rightsquigarrow V_4$	$V_5 \leftarrow\!\!o V_4$	Type 2(a)
	$V_5 o\!\!\rightarrow V_6$	$exp(V_5) \not\rightsquigarrow V_6$	$V_5 \leftrightarrow V_6$	Type 1(a)
$exp(V_4)$	$V_4 o\!\!-\!\!o V_2$	$exp(V_4) \not\rightsquigarrow V_2$	$V_4 \leftarrow\!\!o V_2$	Type 2(a)
	$V_4 o\!\!-\!\!o V_3$	$exp(V_4) \not\rightsquigarrow V_3$	$V_4 \leftarrow\!\!o V_3$	Type 2(a)
	$V_4 o\!\!\rightarrow V_5$	$exp(V_4) \rightsquigarrow V_5$	$V_4 \rightarrow V_5$	Type 1(b)
	$V_4 o\!\!\rightarrow V_6$	$exp(V_4) \rightsquigarrow V_6$	$V_4 \rightarrow V_6$	Type 1(b)
$exp(V_3)$	$V_3 o\!\!-\!\!o V_2$	$exp(V_3) \not\rightsquigarrow V_2$	$V_3 \leftarrow\!\!o V_2$	Type 2(a)
	$V_3 o\!\!\rightarrow V_4$	$exp(V_3) \rightsquigarrow V_4$	$V_3 \rightarrow V_4$	Type 1(b)
$exp(V_2)$	$V_2 o\!\!-\!\!o V_1$	$exp(V_2) \not\rightsquigarrow V_1$	$V_2 \leftarrow\!\!o V_1$	Type 2(a)
	$V_2 o\!\!\rightarrow V_3$	$exp(V_2) \rightsquigarrow V_3$	$V_2 \rightarrow V_3$	Type 1(b)
	$V_2 o\!\!\rightarrow V_4$	$exp(V_2) \mid V_3 \rightsquigarrow V_4$	$V_2 \rightarrow V_4$	Type 1(c)

Table 4. Example steps in disambiguating edges by performing experiments.

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 6(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 6(b).

6 Parametrisation of SMCMs

As mentioned before, in his work on causal inference, Tian provides 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, a parametrisation to efficiently store the JPD over the observables is not provided.

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

6.1 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:

$$P(v) = \sum_{\{l_k | L_k \in L\}} \prod_{V_i \in V} P(v_i | Pa(v_i), LPa(v_i)) \prod_{L_j \in L} P(l_j), \tag{6}$$

where $LPa(v_i)$ are the latent parents of variable V_i and $Pa(v_i)$ are the observable parents of V_i.

Remember that in a SMCM the latent variables are implicitly represented by bi-directed edges, then consider the following definition.

Definition 14. *In a SMCM, the set of observable variables can be partitioned into disjoint groups 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, 2002a).*

E.g. in Figure 6(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 6. Then we can rewrite the JPD as follows:

$$P(v) = \prod_{i \in \{1,\ldots,k\}} Q[S_i] \tag{7}$$

Given a SMCM G and a topological order O, the PR-representation has these properties:
1. The nodes are V, the observable variables of the SMCM.
2. The directed edges that are present in the SMCM are also present in the PR-representation.
3. The bi-directed edges in the SMCM are replaced by a number of directed edges in the following way: 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) except if there was already an edge between nodes V_i and V_j.

Table 5. Obtaining the parametrised representation from a SMCM.

Finally, Tian and Pearl (2002a) 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 $V^{(0)} = \emptyset$.

$$Q[S] = \prod_{V_i \in S} P(v_i | (T_i \cup Pa(T_i)) \backslash \{V_i\}) \tag{8}$$

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 all variables $V \backslash V'$ from the graph 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 that is associated with the SMCM as above.

6.2 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.

PR-representation

Consider $V_{o_1} < \ldots < V_{o_n}$ to be a topological order O over the observable variables V, and let $V^{(i)} = \{V_{o_1}, \ldots, V_{o_i}\}$, $i = 1, \ldots, n$ and $V^{(0)} = \emptyset$. Then Table 5 shows how the parametrised (PR-) representation can be obtained from the original SMCM structure.

What happens is that 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 (8).

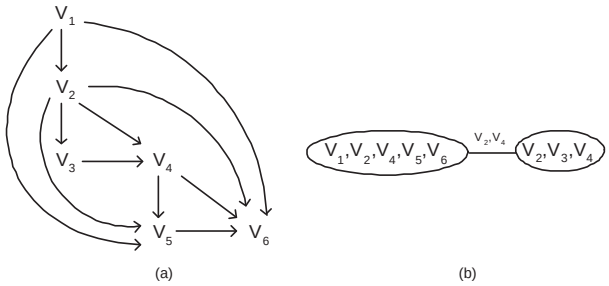

Fig. 9. (a) The PR-representation applied to the SMCM of Figure 6(b). (b) Junction tree representation of the DAG in (a).

In Figure 9(a), the PR-representation of the SMCM in Figure 6(a) can be seen. 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, $V_5 \rightarrow V_6$.

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

Proof. Proving that PR is an I-map of S amounts to proving that all independences represented in PR (A) imply an independence in S (B), or $A \Rightarrow B$. We will prove that assuming both A and $\neg B$ leads to a contradiction.

Assumption $\neg B$: consider that two observable variables X and Y are dependent in the SMCM S conditional on some (possible empty) set of observable variables Z: $X \not\perp\!\!\!\perp_S Y|Z$.

Assumpion A: consider that X and Y are independent in PR conditional on Z: $X \perp\!\!\!\perp_{PR} Y|Z$.

Then based on $X \not\perp\!\!\!\perp_S Y|Z$ we can discriminate two general cases:

1. $\exists$ a path C in S connecting variables X and Y that contains no colliders and no elements of Z.
2. $\exists$ a path C in S connecting variables X and Y that contains at least one collider Z_i that is an element of Z. For the collider there are three possibilities:
   a) $X \ldots C_i \rightarrow Z_i \leftarrow C_j \ldots Y$
   b) $X \ldots C_i \leftrightarrow Z_i \leftarrow C_j \ldots Y$
   c) $X \ldots C_i \leftrightarrow Z_i \leftrightarrow C_j \ldots Y$

   Now we will show that each case implies $\neg A$:

1. Transforming S into PR only adds edges and transforms double-headed edges into single headed edges, hence the path C is still present in S and it still contains no collider. This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false.
2. a) The path C is still present in S together with the collider in Z_i, as it has single headed incoming edges. This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false.
   b) The path C is still present in S. However, the double-headed edge is transformed into a single headed edge. Depending on the topological order there are two possibilities:
      - $C_i \rightarrow Z_i \leftarrow C_j$: in this case the collider is still present in PR, this implies that $X \not\perp\!\!\!\perp_{PR} Y|Z$
      - $C_i \leftarrow Z_i \leftarrow C_j$: in this case the collider is no longer present, but in PR there is the new edge $C_i \leftarrow C_j$ and hence $X \not\perp\!\!\!\perp_{PR} Y|Z$
   c) The path C is still present in S. However, both double-headed edges are transformed into single headed edges. Depending on the topological order there are several possibilities. For the sake of brievity we will only treat a single order here, for the others it can easily be checked that the same holds.
   If the order is $C_i < Z_i < C_j$, the graph becomes $C_i \rightarrow Z_i \rightarrow C_j$, but there are also edges from C_i and Z_i to C_j and its parents $Pa(C_j)$. Thus the collider is no longer present, but the extra edges ensure that $X \not\perp\!\!\!\perp_{PR} Y|Z$.

This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false and therefore we can conclude that PR is always an I-map of S under our assumptions. □

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 (8).

Optimising the Parametrisation

Remark 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, giving precedence to variables with a lesser amount of parents, will cause less edges to be added to the DAG. This is because 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, leading to a more efficient parametrisation.

Note that the choice of the topological order does not influence the correctness of the representation, Theorem 2 shows that it will always be an I-map.

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.

6.3 Probabilistic Inference

Two of the most famous existing probabilistic inference algorithms for models without latent variables are the $\lambda - \pi$ algorithm (Pearl, 1988) for tree-structured BNs, and the *junction tree* algorithm (Lauritzen and Spiegelhalter, 1988) for arbitrary BNs.

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 to its PR-representation which allows us to apply the junction tree (JT) 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 only uses independencies in the DAG, applying it to an I-map of the problem gives correct results. See Figure 9(b) for the junction tree obtained from the parametrised representation in Figure 9(a).

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

6.4 Causal Inference

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

In Spirtes et al. (2000); Zhang (2006), a procedure is discussed that can identify a limited amount of causal inference queries. More precisely only those whose result is equal for all the members of a Markov equivalence class represented by a CPAG.

In Richardson and Spirtes (2003), causal inference in AGs is shown 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 5 and Tian and Pearl (2002a). Thus, if we want to apply Tian's causal inference algorithm, the PR-representation provides all the necessary quantitative information, while the original structure of the SMCM provides the necessary structural information.

7 Conclusions and Perspectives

In this chapter we have introduced techniques for causal graphical modeling with latent variables. We have discussed all classical steps in a modeling process such as learning the structure from observational and experimental data, model parametrisation, probabilistic and causal inference.

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 causal inference 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.

As the experimental learning approach relies on randomized controlled experiments, in general it is not scalable to problems with a large number of variables, due to the associated large number of experiments. Furthermore, it cannot be applied in application areas where such experiments are not feasible due to practical or ethical reasons.

For future work we would like to relax the assumptions made in this chapter. 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 (2005a) after each experiment, might help to reduce the number of experiments needed to fully orient the structure. In this way we could extend our previous results (Meganck et al., 2006) on minimising the total number of experiments in causal models without latent variables, to SMCMs. This allows to compare practical results with the theoretical bounds developed in Eberhardt et al. (2005).

SMCMs have not been parametrised in another way than by the entire joint probability distribution, 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 the size and thus the efficiency

of the PR-representation. We would like to investigate this property in a more formal manner. Finally, we have started implementing the techniques introduced in this chapter into the structure learning package (SLP)[7] of the Bayesian networks toolbox (BNT)[8] for MATLAB.

Acknowledgements

This work was partially funded by a IWT-scholarship. This work was partially supported by the IST Programme of the European Community, under the PASCAL network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

Ali, A. and Richardson, T. (2002). Markov equivalence classes for maximal ancestral graphs. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–9.

Ali, A. R., Richardson, T., Spirtes, P., and Zhang, J. (2005). Orientation rules for constructing markov equivalence classes of maximal ancestral graphs. Technical Report 476, Dept. of Statistics, University of Washington.

Chickering, D. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498.

Cooper, G. F. and Yoo, C. (1999). Causal discovery from a mixture of experimental and observational data. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 116–125.

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

Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the 14th International Conference on Machine Learning*, pages 125–133.

Heckerman, D. (1995). A tutorial on learning with bayesian networks. Technical report, Microsoft Research.

Huang, Y. and Valtorta, M. (2006). Pearl's calculus of intervention is complete. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 217–224.

Jordan, M. I., editor (1998). *Learning in Graphical Models*. MIT Press.

Jordan, M. I., Ghahramani, Z., Jaakkola, T., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

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

[8] http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html

Lauritzen, S. L. and Spiegelhalter, D. J. (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.

Mackay, D. (1999). Introduction to Monte Carlo methods. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 175–204. MIT Press.

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

Murphy, K. P. (2001). Active learning of causal bayes net structure. Technical report, Department of Computer Science, UC Berkeley.

Neapolitan, R. (2003). *Learning Bayesian Networks*. Prentice Hall.

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

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

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

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

Russell, S. J. and Norvig, P., editors (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Shpitser, I. and Pearl, J. (2006). Identification of conditional interventional distributions. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 437–444.

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

Spirtes, P., Meek, C., and Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. In *Computation, Causation, and Discovery*, pages 211–252. AAAI Press, Menlo Park, CA.

Tian, J. (2005). Generating markov equivalent maximal ancestral graphs by single edge replacement. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 591–598.

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

Tian, J. and Pearl, J. (2002b). On the testable implications of causal models with hidden variables. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 519–527.

Tong, S. and Koller, D. (2001). Active learning for structure in bayesian networks. In *Seventeenth International Joint Conference on Artificial Intelligence*.

Zhang, J. (2006). *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD thesis, Carnegie Mellon University.

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

Zhang, J. and Spirtes, P. (2005b). A transformational characterization of markov equivalence for directed acyclic graphs with latent variables. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 667–674.