

# **Incorporation of Latent Variables in Causal Discovery Using Experimental Data**

Honours Thesis – Semester 2 2006

*Christopher D. Fell (ID: 18979246)*

*Supervisor: Kevin B. Korb*

**School of Computer Science and Software Engineering  
Monash University**

**Bachelor of Software Engineering – Honours (2770)  
Clayton Campus**

**Abstract**

Discovering the causal structure behind a body of statistical data is a difficult task. One of the main classes of algorithms used to learn causal structures, known as constraint-based algorithms, currently is only able to find partial data about a causal model. Some arcs are left partially or completely undirected, and very little information is determined about the structure of latent variables affecting the model. This paper examines how much additional information can be determined about the causal structure and the presence of latent variables, through the incorporation of experimental intervention on specific variables. These intervention techniques are then incorporated into the IC-Intervention algorithm, with experimental application of the algorithm giving promising results.

## Contents

1	Introduction . . . . .	4
2	Statistics and Causation . . . . .	5
2.1	Graphical Representations of Causal Models . . . . .	5
2.2	Latent Variables . . . . .	7
3	Causal Discovery . . . . .	7
3.1	Metric-based Discovery Algorithms . . . . .	8
3.1.1	CaMML . . . . .	8
3.1.2	EM method . . . . .	9
4	Constraint-based Discovery Algorithms . . . . .	10
4.1	D-Separation . . . . .	11
4.2	IC algorithm . . . . .	12
4.3	PC Algorithm . . . . .	13
4.3.1	Modified PC Algorithm . . . . .	14
4.4	CI Algorithm . . . . .	15
4.4.1	Partially Oriented Inducing Path Graphs (POIPG) . . . . .	15
4.4.2	CI Algorithm . . . . .	16
4.4.3	FCI Algorithm . . . . .	17
4.5	Intervention and Experimentation . . . . .	18
5	Process . . . . .	19
5.1	Markov-Equivalent Graphs . . . . .	19
5.2	Intervention . . . . .	22
5.3	IC-Intervention Algorithm . . . . .	27
6	Experiments . . . . .	30
6.1	Experimental methods . . . . .	30
6.2	Experimental Results . . . . .	30
6.3	Discussion . . . . .	31
7	Conclusions and Future Work . . . . .	33
A	Tested Networks . . . . .	36
B	Source Code . . . . .	38

## List of Figures

1	Example of Wright's path modelling (from [28]) . . . . .	6
2	Example of a latent variable . . . . .	8
3	Examples of where d-separation may take effect . . . . .	11
4	$\langle E, F, G, A, C, B \rangle$ is a definite discriminating path for C. . . . .	16
5	Simpson's paradox, and how intervention could be used . . . . .	19
6	Markov-equivalent models involving 2 measured variables . . . . .	20
7	Markov-equivalent models involving 3 variables . . . . .	21
8	Markov-equivalent models involving 3 variables (continued) . . . . .	22
9	Intervention variables distinguishing Markov equivalent models . . . . .	23
10	Three-variable models described in Table 2 . . . . .	25

## 1 Introduction

For a long time now science has been attempting to find out more about the world. In the natural sciences, such as physics, chemistry and biology, empirical experimentation and associated theories are able to account for the vast majority of research undertaken. In other fields, however, it is difficult or impossible to use experiments for all our desired studies, whether because of practical or ethical reasons. For practical reasons, it is impossible to conduct experiments to find out how the universe came into being, or experiments about how particular governmental styles affect the running of a country. Ethically, we cannot deliberately infect people with deadly diseases in order to study cures, or put children into poverty to discover what effect it may have on their adult life.

As an alternative to pure experimentation and abstract theorising in such situations, social, behavioural and medical scientists often utilise statistical analysis in their attempts to find out more about how the world works. Often there are huge bodies of data available to such researchers [4, 3, 13], but techniques are still being developed for interpreting that data in some meaningful, useful fashion. Through the use of causal modeling, it is possible to represent the causal structure of a complex situation in a simple, easy to comprehend, and easy to manipulate manner. In order to use these causal models, however, we must first somehow derive their structure. A number of algorithms exist to do this, falling under the main categories of metric-based algorithms and constraint-based algorithms.

These algorithms, however, suffer from significant drawbacks. Constraint-based algorithms in particular frequently result in only partial models that, while able to accurately represent the statistical distribution data, may not actually provide any useful information. Furthermore, very few algorithms have any significant ability to detect the presence of latent variables (variables unmeasured in the original statistical distribution). However, these shortcomings can be addressed to some extent through the use of experimental intervention.

In this paper, we attempt to explore a method of enhancing existing constraint-based algorithms to utilise experimental intervention. Hopefully, this will allow us to recover more of the original causal structure behind a causal model, as well as allow us discover as much information as possible about any latent variables or structures affecting the model.

In section 2, we begin with introducing the basic concepts behind causal modeling and latent variables, as well as an introduction to causal discovery in section 3. In section 4 we provide a more detailed analysis of constraint-based algorithms, as well as an explanation of some of the most important constraint-based algorithms. Section 5 contains an explanation of the process of extending these algorithms to use intervention, as well as a detailed description in section 5.3 of the IC-Intervention algorithm that we developed. Then in section 6 we discuss the experimental application of this algorithm to a number of sample networks, including a detailed discussion of these results and their significance in section 6.3. Finally, in section 7 conclusions are drawn and potential future work arising from this project is outlined.

## 2 Statistics and Causation

Ever since researchers have been recording statistics, they have been looking for ways to interpret those statistics. How much better will the wheat crop be if the weather is good? Does a university education make someone likely to get a better job? Do refugees have a negative effect on the economy?

When studying statistics, it is generally fairly easy to come up with evidence of correlation between variables. What can be more difficult, is determining the exact causal relationships of these correlations. If there is a correlation between a pair of variables  $X$  and  $Y$ , then the value of  $X$  may affect the probable value of  $Y$  (also referred to as  $X$  influences  $Y$ ),  $Y$ 's value influence the value of  $X$ , or there could be a third variable  $Z$  that influences both of them. Generally speaking, most causal discovery research assumes that variables cannot cyclically affect each other. Although such a situation may sometimes seem to be the case, generally the variables under consideration are from different time periods, i.e. variable  $X$  at time  $t-1$  affects variable  $Y$  at time  $t$ , which in turn affects variable  $X$  at time  $t+1$ . Thus, the assumption of no cyclic dependencies amongst causal relationships is generally a valid one. Another assumption normally made in causal modelling is that the model will hold to the 'Markov condition'. This means that no causal influences exist in the system, apart from those that are explicitly represented – unless a correlation is stated to exist between two variables, they cannot directly influence one another.

Many researchers [22, 29] have said that it is impossible to truly determine causation without experimental data, and that observational data will only be able to illustrate correlations in the data. However, while it may be difficult, there do exist a number of methods for analysing causal relationships within a body of data.

One of the major goals of causal analysis is to come up with some kind of accurate, useful causal model. In other words, some kind of model that describes how the variables interact with and affect each other. By using such a model, we are able to make predictions on future events, based on the current state of things; we are able to make decisions based on these predictions; or we can even change the current state of things in order provide some kind of optimal end result. And this is the ultimate goal of statistical analysis – providing information for people that will prove to be useful.

Of course, before we produce a causal model through statistical analysis, we must first decide how to represent such a model. Generally speaking, most causal models today are presented in some kind of graphical format, for ease of interpretation and ease of representation.

### 2.1 Graphical Representations of Causal Models

The first researcher known to have utilised a graphical representation of uncertain causal relationships was Sewall Wright [28], through his technique known as path modeling. This technique has been used frequently in social and behavioural sciences, as it provides a simple linear representation of correlations, and is able to approximately represent most real causal relationships. In path modeling, a variable  $Y$  is related to its parents  $X_1 \dots X_i$  by the following equation:

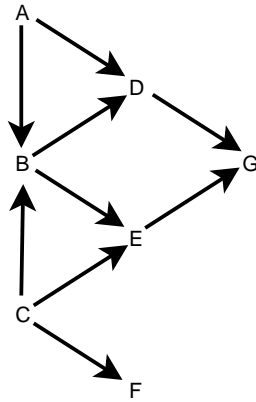


Fig. 1: Example of Wright's path modelling (from [28])

$$Y = P_1X_1 + \dots + P_iX_i + E$$

where  $P_1$  is a constant coefficient, reflecting the degree to which the parent variable  $X_1$  influences  $Y$ , and  $E$  is a latent (unmeasured) error term. An example of this is shown in Figure . Although representing causal influences like this as a linear relationship is rarely precisely accurate, it does allow for a close approximation of many realistic situations [2]. Linear models will be the focus of this project, since they greatly simplify the computations and mathematics of the correlation coefficients, particularly when compared to learning models with discrete values and probability tables.

Two useful concepts to distinguish between when performing causal analysis are the concepts of a *causal model* and a *causal theory*, as described in [26]. A causal model is a directed acyclic graph (DAG), where each node corresponds to a variable, and the directed arcs indicate which variables have a causal influence on which other variables. The graph must be acyclic, since it is impossible for two variables to directly affect each other – as mentioned earlier, in most situations where it appears that two variables affect each other, the reality is that the variables are sequentially separated in time. A causal theory is a causal model, and a set of parameters compatible with that model, which describe the probabilities of each variable relative to its parents. This provides a very convenient way of representing a causal structure, as well as the associated structural equations required to predict the value of any variable, given the values of its parents.

A common way of representing causal theories today is through the use of Bayesian networks [16, 21]. Such networks consist of a DAG, with each node representing a variable, and for each node a conditional probability table, describing the state of the node relative to its parents. Bayesian networks make it easy to update probabilities based on new evidence, as well as a number of software packages, such as Hugin or Netica, being available to make working with Bayesian networks easier. Another advantage of Bayesian networks is that they allow for sequential, time-dependant correlations, through the use of Dynamic Bayesian Networks (DBNs) [20].

Parametrisation of causal theories are generally relatively easy to determine, at least in theory, being simply a matter of statistical estimation and refinement. The practicalities of the process often prove to be difficult, but a number of techniques are offered to perform this task [28, 8]. The more challenging aspect of causal analysis is that of discovering the causal model; or at least, a model that is close to the true model. An example given in Glymour et al's book [12] demonstrated that a model with 6 variables, and 4 possible states of a link between each pair (causation in either direction, no link, or a bi-directional arc indicating spurious correlation), would result in  $4^{15}$  possible causal models. While there would obviously be a lot of these models eliminated due to containing cycles, it is still a huge number, particularly considering the small variable space being considered.

Because the causal modeling problem is obviously so large, there have been a number of algorithms developed to assist with learning causal structure, with varying levels of success. In the following sections, the most important of these algorithms will be discussed.

## 2.2 Latent Variables

Before a discussion of these algorithms, another problem that causal discovery algorithms must account for is that sometimes in our probability distributions, there are variables that we do not have any data for. This can be because it may be physically unmeasurable (e.g. data about the sun from 10 million years ago) or we may not realise that the variable exists to be considered (e.g. abstract concepts such as socioeconomic status). Regardless of whether these variables are unmeasurable or just unmeasured, there are times in the real world where algorithms must be able to deal with variables that are missing from the sampled distribution under consideration. Such unobserved and unknown variables are commonly referred to as 'latent variables'.

The presence of latent variables can have significant effects on a causal model. In Figure 2, a model is given of a suggestion from [1], where rather than the more commonly accepted view of a) smoking directly affecting cancer, the correlation between smoking and cancer is potentially due to b) a particular, unmeasured genotype affecting the probability of both. Statistically, both of these theories can equally represent the statistical data distributions obtained from various studies<sup>1</sup>. This obviously presents a dilemma for researchers, since it could challenge the popular belief that a reduction in smoking would reduce the likelihood of cancer. Thus it falls to causal learning algorithms to attempt to distinguish between these models, and discover as much information as possible about latent variables affect the causal model.

## 3 Causal Discovery

The process of deriving a causal model from statistical data is a challenging one. As mentioned above, there are a huge number of causal structures that can be derived from even a very limited number of variables; when dealing with larger, more complex networks, such as the 37-variable alarm network

---

<sup>1</sup> While the theory of cancer causing smoking also fits the data, common sense and temporal ordering should be able to eliminate some possibilities!

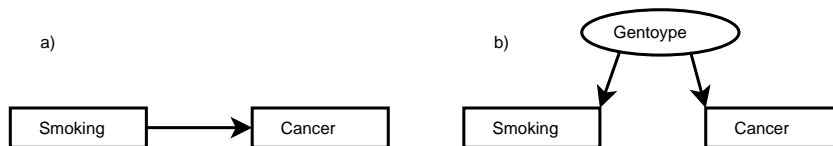


Fig. 2: Example of a latent variable

[3], the number of possible models can become truly overwhelming. There are a number of algorithms that have been developed to deal with this problem. These algorithms can be divided into two major categories – metric-based causal discovery algorithms, and constraint-based causal discovery algorithms.

### 3.1 Metric-based Discovery Algorithms

Metric-based learners, as the name suggests, work on the basis of a metric, which is used to evaluate potential models. The general process of a metric-based learner is to randomly generate a variety of networks, and then score them against some kind of metric – accuracy, complexity, etc. These models are then refined and scored again, and the process is repeated until a set limit has been reached, such as a particular score, or an amount of time passed. Some examples of metric-based learners are discussed below.

#### 3.1.1 CaMML

The CaMML (Causal discovery via MML) program, developed at Monash University [19], is a metric-based causal discovery algorithm, as opposed to constraint-based algorithms such as the IC or PC algorithms (see section 4). CaMML works by generating complete DAGs, and then assessing their suitability for the data they are meant to represent.

CaMML divides the search space into the set of totally ordered models (TOMs), which represent an ordering of the variables. By placing arcs between the variables in a TOM, the arc direction is already implied by the variable ordering, and a DAG can be easily formed. These TOMs are then compared using an MML (Minimum Message Length) metric. According to MML, the ideal model is one that minimises the complexity of the model (minimising the data required to specify the model), and one that also models the sample data as closely as possible (minimising the data required to specify the data compared to the model). which says that In other words, a model which is both simple and closely fits the data, is a good model; trade offs between these two points (complexity and fitness) are what ranks and measures various potential causal models.

The generated TOMs are stochastically sampled – the CaMML program, in it's current form, has implemented a genetic algorithm search and a Metropolis algorithm search. These sampling techniques are intended to eventually determine what is the best TOM, in terms of the MML metric. Note that there are still  $n!$  possible orderings for the TOM, where  $n$  is the number of variables; hence why the TOMs are stochastically sampled as opposed to all possible TOMs being compared.

Once a number of likely TOMs have been identified, they are transformed into DAGs, and grouped if they represent virtually identical DAGS (i.e. the same skeletons and arc directions). These DAGs are compared to the original data, and likely causal models are suggested.

According to experimental evaluation [27], CaMML performs significantly better than TETRAD II at recovering the original network, in that it is capable of recovering the network at much smaller sample sizes. Note that this is not without dispute, as there is no clearly defined or widely accepted metric that provides a practical measure of the quality of causal discovery algorithms. One of the most common metrics, edit distance, has significant drawbacks, for example it regards all arcs as equally important (in measuring how many arcs are different between the causal model output by the algorithm and the original, true model), when obviously some arcs will be far more important than others to the causal model. However, if nothing else, CaMML has an advantage over most constraint-based algorithms, in that it returns a specific DAG for the causal model, as opposed to an equivalence class. While this returned DAG is not necessarily accurate, for some applications the existence of a specific DAG may be important.

### 3.1.2 EM method

Another metric-based algorithm that is able to take into account latent variables is the EM method, originally proposed by Dempster and company [7]. The EM, or Expectation-Maximisation method, randomly selects an initial parametrisation  $\Phi$  for a model and then iteratively improves this parametrisation by repeating two steps:

1. Current parametrisation  $\Phi$  is used to compute the expected values of all statistics in the current structure.
2.  $\Phi$  is replaced by parameters that maximise some kind of scoring metric with the expected statistics.

These steps are repeated until there is no more improvement in the score achieved on the scoring metric. In practical terms, this generally means until some kind of statistically significant level (e.g. 0.5%) is reached, as such a small degree of change is generally not worth considering.

Friedman [10] described a variation on the EM method which could be used to select models, as well as estimate parametrisation. This variation was known as MS-EM, or Model Selection EM. This method works as follows:

1. Choose an initial model  $M^0$  and an initial parametrisation  $\Phi^0$  randomly
2. Loop for  $n = 0, 1, \dots$  until convergence
  - (a) Find a model  $M^{n+1}$  that maximises the expected score
  - (b) Let  $\Phi^{n+1}$  = the parametrisation that maximises the score with the current structure

So, for each stage, the selected model and parameters will attempt to give the highest possible score based on the previous assessment. Of course, in reality it

is not essential to maximise the score, merely to ensure that the score improves. This means that the algorithm will continue iterating, and will keep improving scores up until it reaches some kind of plateau (i.e. a local minimum, maximum or saddle point). The scoring metric used in the paper was Lam and Bacchus' [18] MDL (Minimal Description Length) metric for learning Bayesian networks, which was in turn based on Rissanen's [23] more general work on MDL. MDL is a very similar metric to MML, with the primary difference being that the MML approach more closely follows the original idea of Bayesianism, and so is able to make better use of a prior distribution.

Because the model is randomly chosen, it is a relatively simple matter for the EM method to select a model that includes a latent variable. According to experimental results in [10], introducing latent variables improves the density estimation of the algorithm, but when the number of instances gets larger (more than about 2), the impact of the latent variables is lessened.

A further study by Friedman [11] went on to improve the scoring metric, in an attempt to make it more explicitly Bayesian. According to the experimental results presented in this paper, the more complex Bayesian procedures generally performed better than the simpler, more linear methods, but not always. The difference was especially noticeable in the cases with more latent variables. Friedman believes that this is because the experiments were limited by CPU time, and so the simpler procedures were able to have many more restarts – an important factor, particularly when attempting to search for latent variables.

One major problem with the EM method in general, is that local maxima can easily trap EM in a poor solution. The method keeps searching until the score stops improving, but this may be due to a saddle point, or a local maximum that is significantly lower than the global maximum. An alternative approach that is based on EM is the Information Bottleneck EM algorithm [9], which views the learning problem as a trade-off between two main objectives: compressing information about the training data, and making latent variables informative about the observed attributes, in order to ensure they preserve the relevant information. Through a process of trading off between these two objectives, a high-scoring solution can theoretically be found. This algorithm does have difficulties dealing with a large number of latent variables, as the representation of the joint distribution grows exponentially. Performance-wise, however, it does appear to perform well, in the experimental results provided in [9] showing that it takes about the same amount of time as a standard EM algorithm, while performing better in 94% of cases. In the more complex hierarchical models the performance is worse than most EM runs, but generally speaking it seems to perform quite well, comparatively.

## 4 Constraint-based Discovery Algorithms

As mentioned above, the CaMML program and the EM method are both metric-based causal learners. The design of metric-based learners is such that they will definitely return a DAG, even if that DAG is not necessarily accurate. While they do a 'best-effort' attempt at finding the correct DAG, the accuracy of this result is not definite. While in some applications this is fine, in other areas it is important that all described relations are true, even if not all true relations are described.

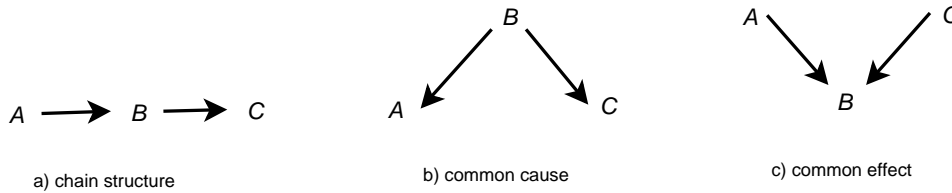


Fig. 3: Examples of where d-separation may take effect

The alternative to metric-based learners that allows this is constraint-based learners – in other words, an algorithm that places constraints on the range of potential causal models, until a minimal-sized (and thus maximally informative) class of potential models is returned. These algorithms attempt to find the set of statistically equivalent models for the causal structure, also referred to as Markov equivalence class. They generally do not return a single DAG, but instead produce what is called a hybrid graph. These hybrid graphs contain the same set of nodes as the causal model, but some arcs may be undirected, and others may be bi-directional. Bi-directional arcs represent a spurious correlation, which indicates the presence of a common-cause latent variable. Undirected arcs will be able to correctly represent all statistical distributions of the model regardless of which way the arc is oriented; the models with the arc oriented in either direction are referred to as Markov equivalent. The set of Markov equivalent models will all contain the same variables as each other, and be capable of representing any of the same probability distributions. The reason why constraint-based algorithms return a Markov equivalence class rather than present a single DAG, is so that they can show what is conclusively known about a given causal model. Below are discussed some of the most important constraint-based algorithms.

## 4.1 D-Separation

First however, a description will be presented of a condition known as d-separation, or directionally-dependent separation. This condition is the basis on which most constraint-based algorithms work.

D-separation occurs when information (or lack thereof) about a certain variable ‘blocks’ a causal path between two other variables. In other words, by knowing the state of the blocking variable, evidence about one side of the block cannot tell us anything about the state of the other side of the block. This is best described with examples – in Figure 3 a number of common situations are shown.

In situation a), the variables  $A$  and  $C$  are d-separated by evidence on variable  $B$ . If we do not know the state of  $B$ , then a change to  $A$  will change  $B$  which will in turn change  $C$ . If on the other hand we know the state of variable  $B$ , then learning the state of  $A$  will tell us nothing more about the state of  $C$ . Further, if we learn something of the state of  $C$ , we cannot use diagnostic reasoning to learn anything new about  $A$ . Intuitively this is obvious, as the ‘blocking’ variable cuts off any causal paths, and thus eliminates any causal influence. A similar situation is b), where once again the variables  $A$  and  $C$

are d-separated by evidence on variable  $B$ . Symbolically, this situation can be written as  $A \perp\!\!\!\perp C \mid \{B\}$ .

Situation c) is slightly different – the variables  $A$  and  $C$  are d-separated by the *lack* of evidence about  $B$ . If we do not know the state of  $B$ , then  $A$  and  $C$  are conditionally independent. If we learn the state of  $B$ , however, information about  $A$  will ‘explain away’ the potential cause of  $C$ , and so these variables have become conditionally dependent. This case is known as a collider and can be written symbolically as  $A \perp\!\!\!\perp C \mid \emptyset$ . The difference between these cases, and the ability to precisely determine the location of a collider through purely statistical analysis, forms the basis of every constraint-based causal learner, starting with the very first example of the IC algorithm.

## 4.2 IC algorithm

The first ever constraint-based causal discovery algorithm was the IC algorithm, originally presented in 1990 by Verma and Pearl [26]. The IC algorithm does not attempt to find an exact DAG or offer a specific causal model fitting the data it was given. Instead, it looks for the class of statistically indistinguishable models that described the data, represented by a *marked hybrid graph*. In such a graph, undirected arcs represent unspecified correlations, bi-directional arcs represent a latent common-cause association, and uni-directional arcs represent causation, either genuine (in the case of marked arcs), or potential (in the case of unmarked arcs).

Because of the way the graph is generated, it produced a class of graphs that contained the true causal model, and in theory all of the graphs within the class would be statistically indistinguishable. Of course, this does not mean that the class of graphs is necessarily very restrictive, nor does it mean that the algorithm is in any way practical or computationally feasible.

The algorithm is described as follows:

Input:  $\hat{P}$  a sampled distribution

Output:  $\mathbf{core}(\hat{P})$  a marked hybrid acyclic graph

1. Beginning with the set of unconnected variables, put an undirected link between every pair of variables  $A - B$ , where there is no set  $S_{AB}$  such that  $(A \perp\!\!\!\perp B \mid S_{AB})$ .
2. For each pair of non-adjacent variables  $A$  and  $B$ , with a common neighbour  $C$ ; if  $C$  is not in any set  $S_{AB}$  from step 1 (i.e. for every  $S$  such that  $A, B \notin S$  and  $C \in S$ ,  $\neg(A \perp\!\!\!\perp B \mid S)$ ), add arrowheads towards  $C$ , forming the structure  $A \rightarrow C \leftarrow B$ .
3. Form  $\mathbf{core}(\hat{P})$  by recursively adding arrowheads according to the rules:
  - If  $A$  is connected to  $B$  and there is a strictly directed path from  $A$  to  $B$ , then direct  $A \rightarrow B$
  - If  $A$  and  $B$  are not adjacent, but either  $A \rightarrow C$  or  $A \leftrightarrow C$ , and  $B - C$ , then direct  $B \rightarrow C$
4. If  $A \rightarrow B$  or  $A \leftrightarrow B$  then mark every uni-directed link  $B \rightarrow C$  in which  $C$  is not adjacent to  $A$ .

This algorithm, like most other constraint-based algorithms, relies on the fact that it is statistically possible to distinguish common-effect structures (also referred to as v-structures or uncovered colliders) where two otherwise unconnected nodes both have a causal influence on a third node; from other similar structures of three nodes such as common-cause or a causal chain.

The IC algorithm however, suffered from a number of practical problems. For one thing, because it involves the examination of every possible subset of variables for each pair of variables, the first step of the algorithm is computationally very expensive for any non-trivial model – in fact, this step of the algorithm is exponential with relation to the number of variables, in both best- and worst-case scenarios.

Further, in order to determine whether two variables are conditionally independent (i.e.  $(A \perp\!\!\!\perp B \mid S_{AB})$ ), the algorithm relies on an all-knowing ‘oracle’, which tells us only of the independencies between the input variables in the given situation. Although the algorithm works in theory, it would be virtually impossible to implement it directly in a program. Other algorithms, however, have been developed in an attempt to make the concept presented in the IC algorithm more practical.

### 4.3 PC Algorithm

In one attempt to address the shortcomings of the IC algorithm, Spirtes, Glymour and Scheines proposed the PC algorithm [25], which was then implemented in the TETRAD program [24]. The PC algorithm uses the same basic principles as the IC algorithm, with a few differences. In the algorithm,  $\mathbf{Adjacencies}(G,A)$  is the set of vertices adjacent to  $A$  in the DAG  $G$ :

1. Form the complete undirected graph  $G$  on the vertex set  $\mathbf{V}$
2. Let  $n = 0$ 
  - repeat
    - repeat
      - select an ordered pair of variables  $A$  and  $B$  that are adjacent in  $G$  such that  $\mathbf{Adjacencies}(G,A) \setminus \{B\}$  has cardinality  $\geq n$ , and a subset  $\mathbf{S}$  of  $\mathbf{Adjacencies}(G,A) \setminus \{B\}$  of cardinality  $n$ , and if  $A$  and  $B$  are d-separated given  $\mathbf{S}$  delete edge  $A - B$  from  $G$  and record  $\mathbf{S}$  in  $\mathbf{Sepset}(A,B)$  and  $\mathbf{Sepset}(B,A)$
      - until all ordered pairs of adjacent variables  $A$  and  $B$  such that  $\mathbf{Adjacencies}(G,A) \setminus \{B\}$  has cardinality  $\geq n$  and all subsets  $\mathbf{S}$  of  $\mathbf{Adjacencies}(G,A) \setminus \{B\}$  of cardinality  $n$  have been tested for d-separation
      - Let  $n = n + 1$
    - until for each order pair of adjacent vertices  $A, B$ ,  $\mathbf{Adjacencies}(G,A) \setminus \{B\}$  is of cardinality  $< n$
3. For each triple of vertices  $A, B$  and  $C$ , such that the pairs  $A, B$  and  $B, C$  are adjacent in  $G$ , but  $A$  and  $C$  are not, then if  $B$  is not in  $\mathbf{Sepset}(A, C)$  direct the arcs inwards, as  $A \rightarrow B \leftarrow C$
4. Redirect any additional edges –

- repeat
  - If  $A \rightarrow B$ ,  $B$  and  $C$  are adjacent,  $A$  and  $C$  are not adjacent, and there is no arrowhead at  $B$ , then orient  $B - C$  as  $B \rightarrow C$
  - If there is a directed path from  $A$  to  $B$ , and an edge between  $A$  and  $B$ , then orient  $A - B$  as  $A \rightarrow B$

until no more edges can be oriented

The first drawback of the IC algorithm – that of being computationally expensive – is addressed in the first and second steps of the PC algorithm. By beginning with a fully populated graph and progressively removing arcs, the algorithm can stop when a particular level of detail is reached, and thus not all subsets of the graph may need to be examined. This can vastly improve performance for sparse graphs, although performance does remain exponential in the worst-case situation of very dense graphs.

The second problem with the IC algorithm – the problem of needing an oracle – is solved in the PC algorithm as well, in a rather simple manner. When it is necessary to determine if two variables are independent, a statistical significance test is applied to the relation between the two given the state of the set of other variables. If the correlation between the two variables is below a given statistically significant level (often 0.05), then the variables are considered independent. Such a situation is referred to as a vanishing partial correlation.

#### 4.3.1 Modified PC Algorithm

Of course, neither of these algorithms tells us much that is useful about latent variables. They both return a *hybrid graph*, meaning that edges may have uni-directional, bi-directional, or undirected arcs. Uni-directional arcs are (at least in theory) a specific one-way causal relationship, and undirected arcs imply that there is some as-yet-undetermined relationship between the variables. Bi-directional arcs mean that both variables have a statistical effect on each other, implying that they are both being affected by another latent variable. Bi-directional arcs, however, do not normally occur in the standard PC algorithm – any such situations are left as undirected arcs.

In [25] however, a modification was suggested to the standard PC algorithm, that would make the search for latent variables simpler. While it would not greatly increase the number of bi-directional arcs, it provided finer differentiation of undirected arcs. With this modified algorithm, arcs could be distinguished between potentially having an arrowhead (thus leading to potential bi-directional arcs and, by implication, latent variables), and specifically having or not having an arrowhead. This requires new notation, as follows: an arc end is represented by  $\rightarrow$  if that endpoint may or may not have an arrowhead, and an arc end represented by  $\rightarrow^*$  may be any type of endpoint ( $\rightarrow$ ,  $-$ , or  $\rightarrow$ ), although this latter notation is only used in the algorithm, not the output. The algorithm uses the same first two steps as the standard PC algorithm, and then continues from step 3 as follows:

1. ...
2. ...

3. Let  $F$  be the resulting graph from step 2. If  $A$  and  $B$  are adjacent in  $F$ , orient the edge between  $A$  and  $B$  as  $A \circ\circ B$ .
4. For each triple of vertices  $A, B$  and  $C$ , such that the pairs  $A, B$  and  $B, C$  are adjacent in  $F$ , but  $A$  and  $C$  are not, then if  $B$  is not in  $\text{Sepset}(A, C)$  direct the arcs inwards, as  $A * \rightarrow B \leftarrow * C$
5. Redirect any additional edges –
  - repeat
    - If  $A * \rightarrow B$ ,  $B$  and  $C$  are adjacent,  $A$  and  $C$  are not adjacent, and there is no arrowhead at  $B$ , then orient  $B * \rightarrow C$  as  $B \rightarrow C$
    - If there is a directed path from  $A$  to  $B$ , and an edge between  $A$  and  $B$ , then orient  $A - B$  as  $A * \rightarrow B$

until no more edges can be oriented

This algorithm, though a slight improvement on the standard PC algorithm, still does not provide a lot of useful information on the potential existence of latent variables. Marginal probability distributions with unmeasured variables sometimes give the appearance of some spurious direct connection between a pair of variables. Also, if we allow bi-directional edges into the output of the algorithm, then some of the assumptions about d-separation given a subset of **Adjacencies** are no longer valid. In order to provide more useful and accurate information about the causal model, major changes are needed to both the procedure and to the interpretation of the output. Such changes are demonstrated in the CI algorithm.

## 4.4 CI Algorithm

### 4.4.1 Partially Oriented Inducing Path Graphs (POIPG)

Inducing paths were first introduced by Verma and Pearl [26] as a method of characterising the conditions under which two variables in a set  $\mathbf{O}$  of observed variables are not d-separated, given any subset of the other variables in  $\mathbf{O}$ .

Assume  $G$  is a DAG over a variable set  $\mathbf{V}$ , and  $\mathbf{O}$  is a subset of  $\mathbf{V}$  containing  $A$  and  $B$ , and  $U$  is an undirected path from  $A$  to  $B$ . Then  $U$  is an inducing path relative to  $\mathbf{O}$  if and only if every member of  $\mathbf{O}$  on  $U$  except for the endpoints is a collider on  $U$ , and also each collider on  $U$  is an ancestor of either  $A$  or  $B$ . Furthermore,  $A$  and  $B$  are not d-separated by any subset  $\mathbf{Z}$  of  $\mathbf{O} \setminus \{A, B\}$  if and only if there is an inducing path over the subset  $\mathbf{O}$  between  $A$  and  $B$ . When such a path is represented in an inducing path graph, the arcs in the graph may be either uni-directional, or if there is a latent common cause of both  $A$  and  $B$  that is in  $\mathbf{V}$  but not in  $\mathbf{O}$ , then the arc will be bi-directional.

In the interests of computability, the CI algorithm attempts to find a partially oriented inducing path graphs (POIPG). Like the output of the modified PC algorithm, a POIPG can contain several kinds of edges, including  $A \rightarrow B$ ,  $A \circ\rightarrow B$ ,  $A \circ\circ B$ , or  $A \leftrightarrow B$ . Consistent with the earlier definitions, a ‘ $\circ\rightarrow$ ’ indicates that it is not known whether or not an arrowhead occurs at that end of the arc, and ‘ $*$ ’ indicates any of an arrow or an empty mark.

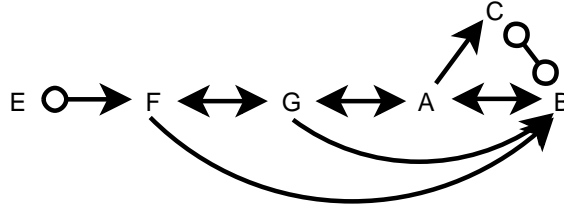


Fig. 4:  $\langle E, F, G, A, C, B \rangle$  is a definite discriminating path for  $C$ .

#### 4.4.2 CI Algorithm

The CI (Causal Inference) algorithm is the initial algorithm capable of producing a POIPG. Once again, it follows the common format for constraint-based algorithms – first, determine the adjacencies between nodes, and then orient those adjacencies as much, and as usefully as possible.

In order to use this algorithm, one must be able to identify a structure known as a ‘definite discriminating path’. In a POIPG,  $U$  is a definite discriminating path for  $M$  if and only if  $U$  is an undirected path between  $A$  and  $B$  containing  $M$ , where  $M$  is neither  $A$  or  $B$ , every vertex on  $U$  except for the endpoints and  $M$  is a collider or a definite non-collider on  $U$ , and

1. if  $V$  and  $V'$  are adjacent on  $U$ , and  $V$  is between  $V$  and  $B$  on  $U$ , then  $V \ast \rightarrow V'$  on  $U$ .
2. if  $V$  is between  $A$  and  $M$  on  $U$  and  $V$  is a collider on  $U$  the  $V \rightarrow B$  in the POIPG, else  $V \leftarrow \ast B$
3. if  $V$  is between  $B$  and  $M$  on  $U$  and  $V$  is a collider on  $U$  the  $V \rightarrow A$  in the POIPG, else  $V \leftarrow \ast A$
4.  $A$  and  $B$  are not adjacent in the POIPG

An example of this is illustrated in Figure 4 (reproduced from [25]). With that in mind, the CI algorithm operates as follows:

1. Form the complete undirected graph  $Q$  on the vertex set  $\mathbf{V}$ .
2. If  $A$  and  $B$  are d-separated given any subset  $\mathbf{S}$  of  $\mathbf{V}$ , delete the edge between  $A$  and  $B$  and record  $\mathbf{S}$  in  $\mathbf{Sepset}(A,B)$  and  $\mathbf{Sepset}(B,A)$
3. Let  $F$  be the graph from step 2. Orient all edges as  $\circ - \circ$ . For each v-structure  $A, B, C$ ; then if  $B$  is in  $\mathbf{Sepset}(A,C)$  orient  $A \ast - \ast B \ast - \ast C$  as  $A \ast \rightarrow B \leftarrow \ast C$ , otherwise orient as  $A \ast - \ast \underline{B} \ast - \ast C$
4. Orient the edges –
  - repeat
    - If there is a directed path from  $A$  to  $B$ , and an edge  $A \ast - \ast B$ , orient  $A \ast \rightarrow B$  to prevent a cycle forming.
    - else if  $B$  is a collider along  $\langle A, B, C \rangle$ ,  $B$  is adjacent to  $D$ , and  $A$  and  $C$  are not d-connected given  $D$ , then orient  $B \ast - \ast D$  as  $B \leftarrow \ast D$

else if  $U$  is a definite discriminating path between  $A$  and  $B$  for  $M$  in the POIPG, and  $P$  and  $R$  are adjacent to  $M$  on  $U$ , and  $P - M - R$  is a triangle,

– then

if  $M$  is in  $\text{Sepset}(A,B)$  then  $M$  is marked as a non-collider on sub-path  $P \text{---} M \text{---} R$

else orient  $P \text{---} M \text{---} R$  as  $P \rightarrow M \leftarrow R$

else if  $A \rightarrow B \text{---} C$  then orient as  $A \rightarrow B \rightarrow C$

until no more edges can be oriented.

The CI algorithm as outlined above is impractical for models with large numbers of variables. This is for two reasons, both related to the way the algorithm determines adjacencies. Firstly, there are too many subsets  $\mathbf{S}$  of  $\mathbf{V}$  on which to test for  $A$  and  $B$ 's conditional independence, making it impractical for the same reason that the IC algorithm was impractical. Secondly, unless the sample sizes are excessively large, then for discrete distributions there are no reliable tests of independence of two variables conditional on a large set of other variables.

#### 4.4.3 FCI Algorithm

An attempt was made to make the CI algorithm more practical for larger numbers of variables, and the result was the FCI, or Fast Casual Inference algorithm.

The FCI starts out by effectively performing the first 3 steps of the PC algorithm, resulting in a graph with most (though not necessarily all) of the incorrect edges removed, and any v-structures (colliders) will have their edges directed appropriately.

After performing those three steps, we then check for each pair of variables, if they are d-separated given any subset  $\mathbf{S}$  of  $\text{Possible-D-SEP}(A,B) \setminus \{A,B\}$  or any subset  $\mathbf{S}$  of  $\text{Possible-D-SEP}(B,A) \setminus \{A,B\}$ , and if so remove the edge and record  $\mathbf{S}$  in the appropriate **Sepsets**. **Possible-D-SEP** is determined by eliminating certain vertices as definitely not d-separating the two variables, and obviously there is a trade-off between reducing the size of **Possible-D-SEP** and the process of eliminating vertices from the **D-SEP** set.

After performing this extra step in eliminating edges, the remaining edges are then oriented exactly as they are in the standard CI algorithm.

This algorithm is computationally efficient, and gives us a very accurate and informative POIPG, which in turn can tell us a lot about the existence of latent variables in the model. Once we have recovered the maximally informative POIPG, we are able to determine a potentially large amount of information about any latent variables within the model. Statistically, some relations between variables are able to be definitively determined as purely direct causal influences, as represented by uni-directional arcs, while others will result in bi-directional arcs, indicating the presence of a latent variable.

However, despite the fact that the FCI algorithm is far more effective than most other constraint-based algorithms at determining causal structure, it is still unable to conclusively determine the exact nature of all causal relationships in the model. The unknown relations will be represented by arcs with either one or two ' $\rightarrow$ ' ends. Furthermore, while the FCI algorithm is capable of determining the presence of some latent variables, others will remain undetected. And

finally, any correlation due to a latent variable that is detected by the algorithm, that has a direct causal influence as well as the latent common cause, will simply be dismissed as solely due to the latent variable. All of these unknown correlations make excellent candidates for experimental intervention.

## 4.5 Intervention and Experimentation

One method of attempting to determine the exact causal model when purely statistical analysis cannot differentiate, is through the use of experimental intervention. Earlier in this paper, it was mentioned that the popularity of statistical analysis was because experiments were sometimes impossible, but equally, after the causal discovery process, intervening experimentally on even a single variable can often help distinguish between two otherwise identical models, and show which is the true model.

All causal discovery algorithms thus far rely on a concept known as *faithfulness*, which is an application of Occam’s razor. This states that if there are two or more possible, equally likely explanations, assume that the simpler one is true. Chickering [5] devised a set of transformational rules that allow any arc within a Bayesian network to be reversed, and still have it represent the same probability distribution. Obviously, this implies that there are a huge number of available statistically correct models, despite there only being one true causal model, which some have taken to mean that there is nothing special about the causal interpretation of a Bayesian network, since many incorrect models can do the exact same thing, statistically speaking. Further, extraneous arcs can be parameterised to have zero influence while still remaining a part of the causal model.

A feature of these transformations that we can use, however, is that all of these transformations either keep the same of links, or add an extra link in order to reverse one of the arc directions. Thus, any model which has been modified using Chickering’s transformations will have either the same or greater complexity than the original. Assuming that we begin with the true causal model, then any more complicated options arising from these transformations can be ignored. Further, we assume that causal models should be ‘faithful’ to reality, meaning that for every causal path on the model, there should in fact be some probabilistic dependency between the variables in reality. This further encourages the notion that where are two or more potential causal models, both of which can correctly represent the probability distribution, then it is assumed that the simpler model is correct.

Most of the time in statistical analysis, this assumption is correct, but Simpson’s paradox (discussed in [14, 17]) shows that in a situation where multiple factors balance out, reality may be unfaithful to the simplest possible model. In Figure 5, a) demonstrates the true causal structure. If, however, the combined negative effect of  $XZ$  and  $ZY$  exactly balanced with the positive effect of  $XY$ , then a causal discovery algorithm would detect no net correlation between  $X$  and  $Y$ , resulting in the structure shown in b).

A number of papers [17, 15] have discussed the idea of experimental intervention, in relation to Simpson’s paradox. In the situation above, they show that if one added an external intervention variable, it would be possible to unequivocally determine what the true causal structure was. In the diagram c), an extra variable is added intervening on  $Y$ , called  $I_Y$ . Once this variable is

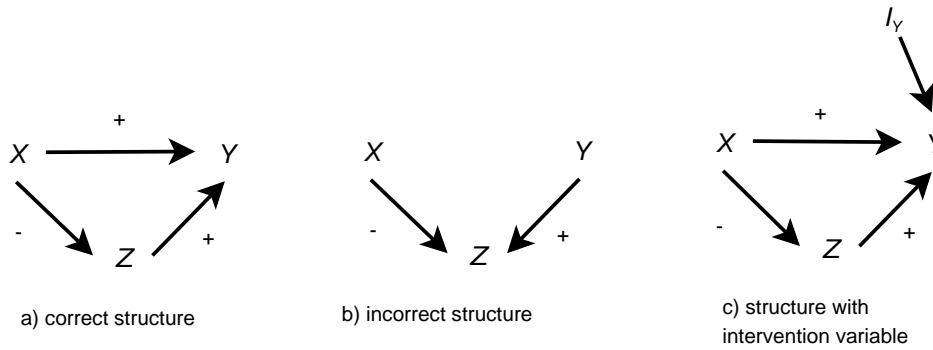


Fig. 5: Simpson's paradox, and how intervention could be used

added, a constraint-based causal discovery algorithm would be able to detect the v-structure of  $X \rightarrow Y \leftarrow I_Y$ , and the true structure of the causal model would be recovered.

Obviously, this is not always possible on all variables, but generally only one of the variables needs to be intervened upon in order to resolve uncertainties.

Another paper by Cooper and Yoo [6] also demonstrated that the use of experimental data in Bayesian causal analysis techniques produced more reliable results than just using observational data. The use of experimental intervention did not significantly affect the causal discovery techniques used – the process of causal discovery using a mixture of experimental and observational data was similar to the process for observational data alone, and the scoring metric that they used differed only in terms of the numerical counts. Though their paper was very limited in its scope of testing, it does seem to also indicate that experimental intervention can be of great use for causal discovery techniques.

## 5 Process

### 5.1 Markov-Equivalent Graphs

As has been mentioned previously, constraint-based causal discovery algorithms generally do not return a true DAG. Instead, they return a graph indicating the class of Markov-equivalent (also called statistically equivalent) graphs into which the DAG falls. Markov-equivalent graphs are graphs which contain the same variables, and any probability distribution which can be represented on one graph can be represented on the other. Because of this, they cannot be distinguished from one another through purely statistical means, even though only one of the Markov-equivalent class is the specific DAG of the true causal model.

Within each Markov-equivalent pattern, there exist a number of possible graphs. For patterns with only two variables, any probability distribution by all patterns with a relationship between the variables, making it impossible to distinguish anything more than the fact that there is a relation. There are no v-structures, nor are there any of the more complex structures used to determine arc orientation. Thus, when latent variables are not included in the graph,

the arc between a pair of variables  $A$  and  $B$  may be directed as either  $A \rightarrow B$  or  $A \leftarrow B$ , while still modeling the correct probability distribution. If a latent variable is included, and influences both of the measured variables (a common-cause latent variable), then as well as the latent variable there may also be an arc in either direction, or no arc at all. See Figure 6 for an illustration of each of these graphs.

Not illustrated is the pattern where the two variables remain unconnected - obviously, in such a situation there is no correlation of any kind between the variables, and thus there is nothing useful that can be learnt. Also not illustrated is the situation where a latent variable affects only one of the measured variables, e.g.  $L \rightarrow A$ . In such a situation, the effect of the latent variable  $L$  will be completely absorbed into the probability distribution of  $A$ . Further, such latent variables are rarely of major concern to researchers, since there may easily be a large number of latent variables which only affect a single variable in the model, and thus do not affect the structure of the causal model at all. In fact, Wright's work on path modeling [28] assumed that each measured variable was affected by a latent 'error' term, in order to represent the randomness of causal modeling.

When patterns with three variables are considered, the range of possible graphs is expanded significantly. The existence of three variables means that a v-structure (collider) may form, which is the only statistically distinguishing feature of a model other than its skeleton [26]. In Figure 7 Pattern 1, the models which contain a single uncovered collider are shown. Figure 7 Pattern 2 are the models with a v-skeleton, but without a collider. Note that in order to avoid forming an uncovered collider and thus changing the Markov equivalence class, only one latent variable may be introduced to the models in Pattern 2, and when it is, the other arc (without a latent variable) may not point into  $A$ .

The final situation with three variables is illustrated in Figure 8 Pattern 3, where all of the variables are connected. In this situation, all models are Markov equivalent, thus each arc may take any of the forms from Figure 6. Because the pattern is fully connected, there are no uncovered colliders, thus arcs may be oriented in almost any direction. The only restriction on the models that are possible, is that a cycle must not be formed. Since this results in literally hundreds of possible permutations and combinations, only a few examples are illustrated in Figure 8.

Again, when there are three variables under consideration it is possible for

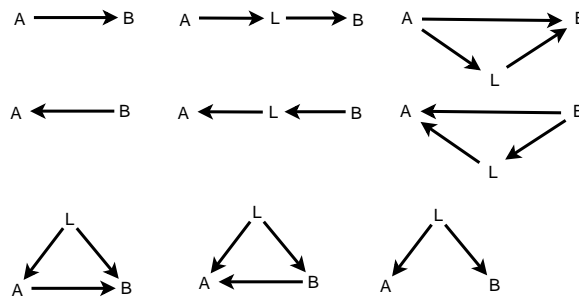
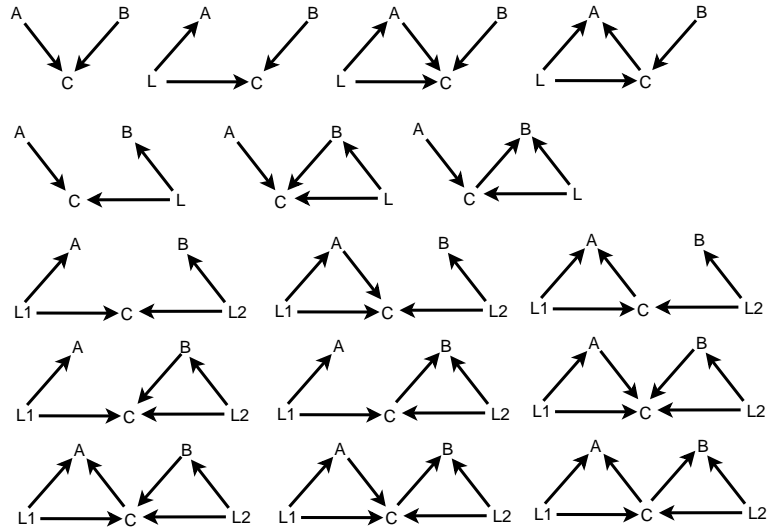
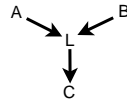


Fig. 6: Markov-equivalent models involving 2 measured variables

**Pattern 1)**



All of these models may have a (possibly additional) latent variable like so, as long as a cycle is not formed:



**Pattern 2)**

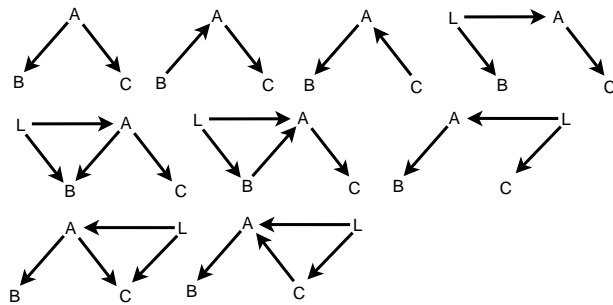


Fig. 7: Markov-equivalent models involving 3 variables

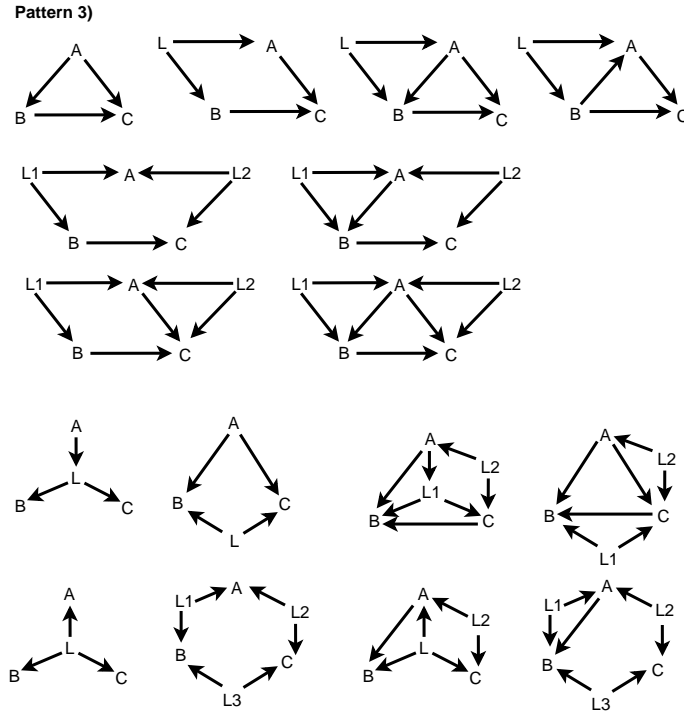


Fig. 8: Markov-equivalent models involving 3 variables (continued)

there to be no interaction between the variables, or for only two of the variables to interact. In the former case, it should once again be obvious that there will be no causal relationships between the variables, and the graph will be completely unconnected. In the latter situation, the two interacting variables will be connected in the same form as one of the potential models outlined above in Figure 6, with the third variable being unconnected.

Also, not discussed in this section is the possibility of a latent variable affecting only a single measured variable, in other words  $L \rightarrow A$ . In this situation, although  $L$  will affect the value that  $A$  has, we cannot learn anything useful about the state of  $A$  from knowing the existence of  $L$ . The variation in the state of  $A$  introduced by the latent variable will be absorbed into  $A$ 's probability distribution.

## 5.2 Intervention

Many researchers ([26, 12, 25]) feel that because a purely statistical analysis of data cannot give us any more detail about the causal model than the Markov equivalence class, then we should not be concerned with trying to find the exact causal model. While it is true that knowing the pattern and the coefficients of the probability distribution allow us to perform some statistical analysis and prediction, it does not tell us the whole picture. The ultimate goal of virtually any research is to discover what we can do to change and improve things. If we define the probability distribution, we can predict results, but if we determine

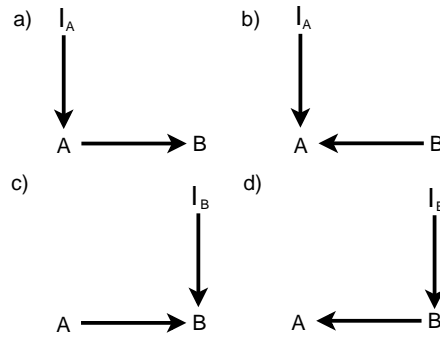


Fig. 9: Intervention variables distinguishing Markov equivalent models

the true causal nature of the model, then we can actively affect the variables.




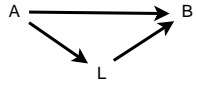
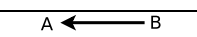
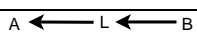
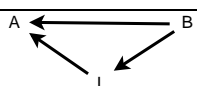
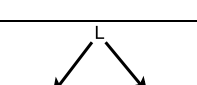
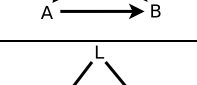
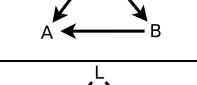
There are two major methods for distinguishing between Markov equivalent models. One of the methods is to assign prior probabilities to various models, although this method is more often used in metric-based causal discovery methods (such as MML) than constraint-based methods (such as the PC or CI algorithms). Either way, the incorporation of prior probabilities requires input from domain experts to identify which models are more probable, and the most probable model is selected from all of the correct, equivalent models. Obviously, this can lead to practical issues, where the cost of an expert analysis may be prohibitively high.

The other method for distinguishing models is by performing experimental intervention. Korb and Nyberg [17] demonstrated that experimental intervention can be used to distinguish models without latent variables, by adding in one or more ‘intervention’ variables, each influencing a measured variable (i.e.  $I_A \rightarrow A$ ). This intervention variable  $I_A$  will force  $A$  to take on a certain value. It is then possible to examine whether a collider forms at  $A$  between  $I_A$  and another variable  $B$ , and thus discover the orientation of the arc between  $A$  and  $B$ . In Figure 9, the use of an intervention variable is demonstrated for a simple case (two measured variables, no latent variables). As can be seen in 9 b) and 9 c), a collider is formed at the intervened-upon variable, allowing us to determine the orientation of the arc between  $A$  and  $B$ . Conversely, in 9 a) and 9 d), we could orient the arc because we know that a collider did not form.

It is possible to examine the status of conditional dependencies/independencies to determine the exact causal model, including all significant latent variables. In Table 1, the states of these dependencies are specified with various models. The notation  $A \perp\!\!\!\perp B \mid \{C\}$  means “ $A$  is conditionally independent of  $B$ , given the set of variables  $\{C\}$ ”.

According to this table, models b), c) and d), as well as models e), f) and g), are indistinguishable even with experimental intervention. However, this indistinguishability is not significant, as any effect that the latent variable has on  $B$  in c) or d) (or the effect on  $A$  in f) or g)) will be absorbed into the probability distribution of  $B$ .

As for the effect of latent variables on the dependencies, it should be apparent from the table that if both  $I_A \not\perp\!\!\!\perp B \mid \{A\}$  and  $I_B \not\perp\!\!\!\perp A \mid \{B\}$ , then it implies that a latent variable exists. Furthermore, if exactly one of either  $I_A \not\perp\!\!\!\perp B \mid \emptyset$  or

	Model	$I_A \perp\!\!\!\perp B \mid \emptyset$	$I_A \perp\!\!\!\perp B \mid \{A\}$	$I_B \perp\!\!\!\perp A \mid \emptyset$	$I_B \perp\!\!\!\perp A \mid \{B\}$
a)		True	True	True	True
b)		False	True	True	False
c)		False	True	True	False
d)		False	True	True	False
e)		True	False	False	True
f)		True	False	False	True
g)		True	False	False	True
h)		False	False	True	False
i)		True	False	False	False
j)		True	False	True	False

Tab. 1: Conditional dependencies among two variables

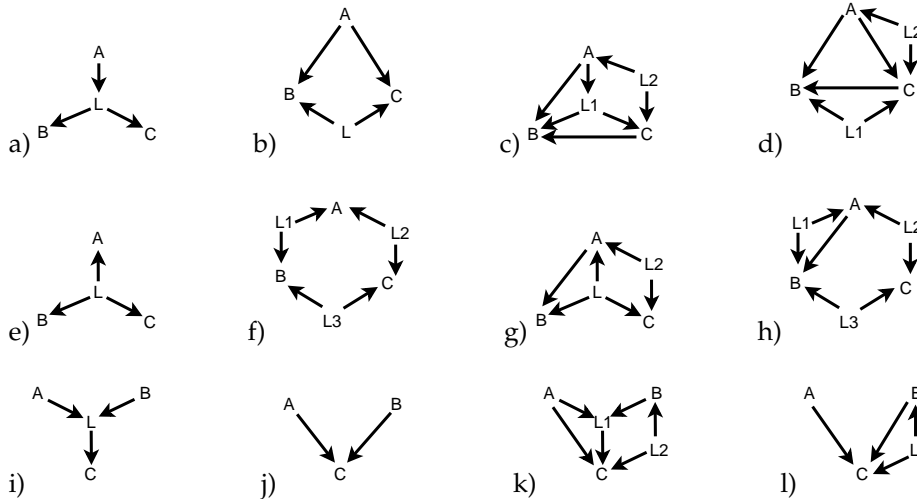


Fig. 10: Three-variable models described in Table 2

$I_{B \not\ll A} \mid \emptyset$  is true, then there also exists a direct causal effect from  $A$  to  $B$  or from  $B$  to  $A$ , respectively. Finally, the only time when both  $I_{A \not\ll B} \mid \emptyset$  and  $I_{B \not\ll A} \mid \emptyset$  are true, is when either there is no correlation at all between the variables, or when there is a latent variable correlating them but no direct causal effect from one to the other.

Worth noting is the fact that even though Table 1 assumes that both variables will be intervened upon, in many situations it is possible to obtain useful extra information about the causal structure even with only one intervention. For example, if  $I_{A \not\ll B} \mid \{A\}$ , then the possible causal model has been narrowed down to either  $A \leftarrow B$ , or a latent variable affecting both  $A$  and  $B$  (though the existence of a latent variable does not preclude a direct correlation as well). However, in order to determine the complete causal structure, generally both variables need to be intervened upon, and thus our studies will be focusing on the case of both variables being intervened upon. The only exception to this, where the complete structure may be determined with only one intervention, is models b), c) and d) above where the intervention occurs on variable  $A$  (or conversely, models e), f) and g) when variable  $B$  is intervened upon). And as has already been mentioned, even with full intervention these models are indistinguishable.

When the process is expanded to three variables, the situation becomes slightly more complex. For the most part, the interactions between three variables  $A$ ,  $B$ , and  $C$  can be modelled as three two-variable interactions,  $A - B$ ,  $B - C$ , and  $A - C$ . When the three-variable situation is viewed as such, the interactions can be determined following the rules set out in Table 2.

The only circumstances where a model (of three variables) cannot be derived solely from examining the pairs of interactions are reflected in Figure 10 – in other words, models where a single latent variable affects all three variables. Potential models of this type can be examined by selecting any triplets which, after examining the pairs of interactions, match any of Figure 10 b), 10 f) or 10 j). These are the situations where assuming only interactions between two

Model	$I_A \perp\!\!\!\perp B \mid \emptyset$	$I_A \perp\!\!\!\perp B \mid \{A\}$	$I_A \perp\!\!\!\perp C \mid \emptyset$	$I_A \perp\!\!\!\perp C \mid \{A\}$
a)	False	True	False	True
b)	False	True	False	True
c)	False	True	False	False
d)	False	True	False	False
e)	True	False	True	False
f)	True	False	True	False
g)	False	False	True	False
h)	False	False	True	False
i)	True	True	False	True
j)	True	True	False	True
k)	True	True	False	True
l)	True	True	False	True

Model	$I_B \perp\!\!\!\perp A \mid \emptyset$	$I_B \perp\!\!\!\perp A \mid \{B\}$	$I_B \perp\!\!\!\perp C \mid \emptyset$	$I_B \perp\!\!\!\perp C \mid \{B\}$
a)	True	False	True	False
b)	True	False	True	False
c)	True	False	True	False
d)	True	False	True	False
e)	True	False	True	False
f)	True	False	True	False
g)	True	False	True	False
h)	True	False	True	False
i)	True	True	False	True
j)	True	True	False	True
k)	True	True	False	False
l)	True	True	False	False

Model	$I_C \perp\!\!\!\perp A \mid \emptyset$	$I_C \perp\!\!\!\perp A \mid \{C\}$	$I_C \perp\!\!\!\perp B \mid \emptyset$	$I_C \perp\!\!\!\perp B \mid \{C\}$
a)	True	False	True	False
b)	True	False	True	False
c)	True	False	False	False
d)	True	False	False	False
e)	True	False	True	False
f)	True	False	True	False
g)	True	False	True	False
h)	True	False	True	False
i)	True	False	True	False
j)	True	False	True	False
k)	True	False	True	False
l)	True	False	True	False

Tab. 2: Conditional dependencies among three variables

variables may incorrectly identify the model. The models in Figure 10 (and the matching dependencies in Table 2) illustrate the basic situations of latent variables affecting three measured variables, the statistically matching situation where latent variables may only affect two variables, and an example of the same with an extra causal arc and latent variable added.

Unfortunately, as can be seen from Table 2, even with experimental intervention it appears impossible to accurately differentiate models of three or more variables, when using only d-separation. Although Glymour [12, pp 247-255] described a method for determining structure of latent variables in more than three variables, his method relied on examining subsets of exactly four variables, and then studying the vanishing tetrad differences between the variables, which is a more complex process than simply determining whether the variables are d-separated. Further, this method was still only capable of determining very limited data about the equivalence class of the latent variable structure, as opposed to determining the exact causal model.

### 5.3 IC-Intervention Algorithm

With the above-discussed set of possible combinations of dependencies and independencies, it becomes a relatively easy task to utilise intervention for discovering the complete causal structure of a network. Although it is possible to only utilise interventions to examine each pair of variables and determine the causal relations between them, the use of experimental intervention in any practical situation does presents certain challenges or difficulties. Because of this, the first step in the IC-Intervention algorithm is to use the basic IC algorithm on the list of variables to resolve as much of the graph as is possible purely through analysis of the statistical data. The use of this algorithm will result in a graph with all adjacencies discovered and represented by arcs, and some of those arcs may be partially or fully directed. Any arc with one or both of its ends represented by a circle  $\circ$ , however, is undirected and must be examined in the later steps of the IC-Intervention algorithm.

This graph is then examined for arcs which have not yet been fully oriented – these pairs of variables are marked as candidates for intervention. In order to improve efficiency and reduce the need for intervention, firstly all cases of  $A \circ \rightarrow B$  (i.e. where only one end of the arc has been oriented) are examined. If variable  $A$  is intervenable, then intervene with a new variable  $I_A$ . If  $I_A \perp\!\!\!\perp B \mid \{A\}$  then this confirms the orientation as  $A \rightarrow B$ ; otherwise, there must be a latent variable acting as a common cause to both  $A$  and  $B$ , as in the case  $A \leftarrow L \rightarrow B$ . In this case, because the latent variable can account for any statistical correlation between  $A$  and  $B$ , the directed arc  $A \rightarrow B$  must be removed. However, there may be a direct causal influence from one variable to another, as well as the correlating latent variable. In order to determine if this is the case, models h), i) and j) from Table 1 must be examined. If  $I_A \not\perp\!\!\!\perp B \mid \emptyset$ , then a direct influence can be oriented as  $A \rightarrow B$  (as well as the latent variable); otherwise variable  $B$  must also be intervened upon. If  $I_B \not\perp\!\!\!\perp A \mid \emptyset$ , then a direct influence is present, but is instead oriented as  $A \leftarrow B$ . Only if both  $I_A \perp\!\!\!\perp B \mid \emptyset$  and  $I_B \perp\!\!\!\perp A \mid \emptyset$ , does it become certain that there is no direct influence, and that all correlation is due to latent variables.

Obviously, this case of  $A \circ \rightarrow B$  needs to be repeated for every pair of similarly directed variables (including  $A \leftarrow \circ B$ ). After this step, the only

---

**Algorithm 1** The IC-Intervention algorithm
 

---

1. Perform complete IC algorithm over the variable space
  2. For each arc left partially directed ( $A \circ \rightarrow B$ ) in the POIPG add an intervention variable  $I_A$  on variable  $A$ .
    - If  $I_A \perp\!\!\!\perp B \mid \{A\}$ , orient the arc as  $A \rightarrow B$
    - Else remove the arc  $A \circ \rightarrow B$  and add a latent common-cause variable affecting both  $A$  and  $B$ . Then,
      - If  $I_A \not\perp\!\!\!\perp B \mid \emptyset$ , add a directed arc  $A \rightarrow B$
      - Else add an intervention variable  $I_B$  on variable  $B$ . If  $I_B \not\perp\!\!\!\perp A \mid \emptyset$ , add a directed arc  $A \leftarrow B$
      - Else do not add a direct arc between  $A$  and  $B$  – they are only correlated by the latent variable
  3. For each arc left completely undirected ( $A \circ \circ B$ ) in the POIPG add an intervention variable  $I_A$  on variable  $A$ .
    - If  $I_A \perp\!\!\!\perp B \mid \{A\}$ , orient the arc as  $A \rightarrow B$
    - Else add an intervention variable  $I_B$  on variable  $B$ . If  $I_B \perp\!\!\!\perp A \mid \{B\}$ , orient the arc as  $A \leftarrow B$
    - Else remove the arc between  $A$  and  $B$  and add a latent common-cause variable affecting both  $A$  and  $B$ . Then,
      - If  $I_A \not\perp\!\!\!\perp B \mid \emptyset$ , add another directed arc  $A \rightarrow B$
      - Else if  $I_B \not\perp\!\!\!\perp A \mid \emptyset$ , add another directed arc  $B \rightarrow A$
      - Else do not add a direct arc between  $A$  and  $B$  – they are only correlated by the latent variable
  4. For each bi-directional arc ( $A \leftrightarrow B$ ), remove the arc between  $A$  and  $B$  and add a latent common-cause variable affecting both  $A$  and  $B$ . Then,
    - Add an intervention variable  $I_A$  on variable  $A$ . If  $I_A \not\perp\!\!\!\perp B \mid \emptyset$ , add a directed arc  $A \rightarrow B$
    - Else add an intervention variable  $I_B$  on variable  $B$ . If  $I_B \not\perp\!\!\!\perp A \mid \emptyset$ , add a directed arc  $A \leftarrow B$
    - Else do not add a direct arc between  $A$  and  $B$  – they are only correlated by the latent variable
-

uncertainty remaining in the graph will be arcs that are completely undirected ( $A - B$ ), and arcs that have an arrowhead at both ends ( $A \leftrightarrow B$ ). The latter of these two cases is slightly simpler to deal with – if such an arc has appeared, then the only consistent structure is that there is a latent variable affecting both  $A$  and  $B$ . Thus, the case of  $I_A \not\perp B \mid \{A\}$  above is repeated, and  $A$  and  $B$  must be intervened upon in turn. If  $I_A \not\perp B \mid \emptyset$ , then a direct influence exists as well as the latent variable, and should be oriented as  $A \rightarrow B$ , or vice-versa if  $I_B \not\perp A \mid \emptyset$ . If, on the other hand, both  $I_A \perp B \mid \emptyset$  and  $I_B \perp A \mid \emptyset$ , then there is no direct arc between  $A$  and  $B$ , only the correlation due to the latent variable affecting both of them. Note that obviously, if  $I_A \not\perp B \mid \emptyset$ , then it is not necessary to also intervene on  $B$ , as the nature of the direct correlation has already been determined.

With an arc that is completely undirected, the first step is to intervene on variable  $A$ . If  $I_A \perp B \mid \{A\}$ , then no latent variable is present and the arc may be safely directed as  $A \rightarrow B$ . Otherwise, there is potentially a latent variable, and so  $B$  must be intervened on as well. For simplicity, only one variable should be intervened on at a time, and thus while the intervention variable  $I_A$  is present, the truth of  $I_A \not\perp B \mid \emptyset$  should be examined as well. If it is the case, the direct influence exists from  $A$  to  $B$ , and so  $B$  must be intervened upon only in order to determine the presence of a latent variable, or lack thereof. Thus, if  $I_A \not\perp B \mid \emptyset$ , then the next step is to intervene on  $B$ , and determine if  $I_B \not\perp A \mid \{B\}$ , and thus determine if a common-cause latent variable is present as well as the direct arc  $A \rightarrow B$ . If on the other hand,  $I_A \perp B \mid \emptyset$ , then the intervention on  $B$  must check both if  $I_B \not\perp A \mid \{B\}$  as before, but also if  $I_B \not\perp A \mid \emptyset$ , in which case there will be both a common-cause latent variable and a direct arc  $A \leftarrow B$ .

Although the first step of the IC-Intervention algorithm calls for the use of the IC algorithm, any algorithm which outputs a POIPG would be equally compatible with this step. Steps two through four of the IC-Intervention merely rely on having an accurate POIPG as input, not a maximally informative one. Other algorithms which output a POIPG, such as the modified PC algorithm or the FCI algorithm from [25], could also work for the first step, or any future constraint-based algorithm which output a similar graph could be utilised. The reason why the IC algorithm was chosen was that it is simple to implement, and produces a hybrid graph with all of the obvious arcs oriented. Given that the IC-Intervention algorithm will (in theory) recover all the information about all direct connections in the network, as well as discover the presence of latent variables wherever they affect a correlation, the only reason to desire a more informative POIPG is to reduce the number of interventions required for this algorithm.

Also, not included as part of this algorithm is the ability to mark the potential presence of any more complex latent structures. As was noted in section 5.2, it appears impossible to tell purely through conditional independencies whether a latent variable interacts with more than two measured variables. However, it would be possible to note any parts of the graph that may be due to a latent variable affecting three (or more) measured variables. This would simply require examining each triplet of variables, and determining whether they match up to one of the cases shown in Figures 10 b), f) or j). These triplets may contain only these correlations, or they may contain any number of other correlations – as shown in 10 d), h) and l), even with extra arcs or latent variables added, the model may still arise either from separate interactions or from a single latent variable affecting all three measured variables. The

important factor is that it contains the latent common-cause correlations and the direct correlations listed in Figures 10 b), f) or j). Additionally, this step could be extended to structures of more than three variables; however, such a process is beyond the scope of this project.

## 6 Experiments

### 6.1 Experimental methods

For testing purposes, the algorithm was implemented using Java 5, and tested on a variety of networks running on an AMD64 3000+ workstation with 1GiB of RAM. For the first step of the IC-Intervention learner, the program ran a modified version of the IC algorithm, as described in [25, pp 166]. The only significant difference to the original IC algorithm is that when a collider is found, the arcs are oriented as  $A^* \rightarrow B \leftarrow *C$ , as opposed to  $A \rightarrow B \leftarrow C$ ; and that when all the adjacencies are recovered the arcs are initialised as  $\circ-\circ$ , instead of leaving them undirected. This is to differentiate arcs which have been determined to be a parent or child, and arcs which are still indeterminate, as an arc with an arrowhead at one end may still have an arrowhead at the other if a latent common-cause is present. Although implementing the full FCI algorithm may have provided a more detailed input for the second step of the IC-Intervention algorithm, due to time constraints only the IC algorithm was used. As mentioned in section 5.3, the use of the IC algorithm would not affect the final effectiveness of the IC-Intervention algorithm at all, only its efficiency and reliance on adding intervention nodes.

The program began with the true causal model read in from a Netica file (DNE file), and after optionally marking some of the nodes in the complete network as latent variables, began the process of recovering the causal structure using the IC-Intervention algorithm. Conditional dependencies and independencies were determined through the use of an “independence oracle”, thus eliminating any issues arising from random ‘noise’ in the statistical data. The recovered network was output after the IC algorithm step for comparison purposes, to see how much of the network remained undiscovered before intervention. The recovered was output again after the entire algorithm had been completed.

The algorithm was tested on a variety of networks (see Appendix A for the full list). This selection of networks was chosen in order to test both dense and sparse networks, large and small networks. Additionally, the tested networks were also chosen to demonstrate a range from zero to many latent variables in a variety of latent structures. It should be emphasised that the networks were not chosen to favour the results of the IC-Intervention algorithm; the only factors considered were the need to demonstrate a variety of structures while still being relatively easy to implement the models.

### 6.2 Experimental Results

The results for each of the tested networks may be found in appendix A, Table 4, along with the list of networks tested. The results indicate that for all of the tested networks, the IC-Intervention algorithm was able to recover

all direct correlations, as well as determining the presence of latent variables affecting a correlation. However, for more complex latent structures than a single common-cause latent variable affecting two measured variables, such as network e), the algorithm was unable to determine anything more than the fact that some latent structure was present.

In all of the networks tested, some degree of intervention was required to completely recover the causal structure. In other words, the IC algorithm always left at least some arcs partially or completely undirected. This is due to the fact that some arcs will have to be root nodes, and because of the nature of constraint-based learners, they will never be able to tell whether the influence from a root node is direct or if it is due to a latent variable. This is illustrated in network 4 b) – the IC algorithm (or any other constraint-based algorithm) is unable to distinguish between the influences on cancer from ‘Smoking’ and ‘Pollution’.

Efficiency with the algorithm does not appear to be a significant issue. Even when run on the 37-node ALARM network (from [3]), the total running time of the algorithm once adjacencies had been recovered (i.e. after the first step of the IC algorithm) was only approximately four seconds. The intervention part of this was completed in less than a second. Given that the recovery of the adjacencies for this network took more than ten minutes, the IC-Intervention algorithm appears to have minimal time requirements for recovering the network. Certainly, for all more trivial networks tested, the algorithm completed virtually instantaneously. Given that one of the main improvements of the FCI over the IC algorithm was an increase in efficiency of finding adjacencies, then if this program were implemented with the FCI algorithm as opposed to the IC algorithm, it would likely prove very efficient indeed, even for very large networks.

### 6.3 Discussion

As can be seen in the results described above, in all of the tested cases the orientation of all direct arcs within the causal networks were able to be determined through the use of the IC-Intervention algorithm. Further, the algorithm is also capable of determining which correlations that are partly or completely due to latent variables. This is compared to most other causal learning techniques which are unable to even determine all correlations that are due to latent variables, and even when they do detect the presence of a latent variable, it is generally impossible to also detect whether there is a direct correlation as well as the latent correlation. Also considering that the IC-Intervention algorithm is both conceptually and algorithmically very simple, as is indicated in section 6.2 by the minimal processing time required, it appears that the IC-Intervention algorithm is a very powerful tool for the recovery of causal networks.

However, the algorithm is not perfect – it is only able to determine whether or not an individual interaction is affected by a latent variable. As described in section 5.2, the algorithm is unable to determine the exact latent structure when a latent variable affects more than three measured variables. Furthermore, the algorithm is only able to determine that there is some sort of latent structure affecting a given correlation. In situations where there is a more complex latent structure than a single latent variable affecting two measured variables, such as the network shown in appendix A network c), the algorithm will only

detect the fact that the correlation between nodes ‘Dyspnoea’ and ‘X-ray’ is being affected by a latent structure. It is undoubtedly useful to know of the presence of latent variables, and the IC-Intervention algorithm does highlight correlations affected by latency, allowing for further study and investigation. Also, the algorithm could be expanded in the future to highlight any structures which may be the result of three (or more) measured variables being correlated by a latent variable, such as those structures highlighted in Table 2. This would involve taking the measures described in section 5.3, of going through the final output and highlighting any groups of variables whose structure may have arisen through the presence of a complex latent structure.

Another issue with the algorithm as implemented for testing, is that the program utilises an “independence oracle”, in order to simplify the programming and reduce additional external sources of error such as random noise in the sample data. However, any instance where the algorithm would be used in reality would not have access to such an oracle; instead some kind of statistical significance test would have to be used. While this would not necessarily add significantly to the complexity of the algorithm, it would make the result susceptible to any random variation, or ‘noise’ in the sample data. Admittedly, this problem affects all statistical analysis techniques, but nevertheless it is an extra source of error in the causal discovery process. While this issue does not affect the validity of any results obtained, it is a factor that would need to be accounted for before the algorithm could be put to any sort of practical use.

Also, it should be pointed out that although the algorithm glosses over the actual process of experimental intervention, adding interventions to a model may carry certain practical difficulties. As was mentioned in the introduction, one of the major reasons for using statistical analysis to perform causal discovery, is because one is often unable to intervene on certain variables, due to either practical difficulties, or ethical issues with experimenting upon the subjects in the manner required (e.g. putting children into poverty, or deliberately infecting people with dangerous diseases). And even if it is possible to experimentally intervene on the target variable, there is still a cost (time and/or financial) associated with setting up the intervention, which may make the practical application of this algorithm prohibitive.

Nevertheless, where intervention is possible, this algorithm is capable of recovering more of variable structure than would otherwise be possible. Also, because the input to the second step of the IC-Intervention algorithm simply requires a POIPG, and algorithm that returns such a graph (such as the PC, CI or FCI algorithms, or any other, more effective algorithms that may be developed in the future) could be used in the first step of the IC-Intervention algorithm, to greater or lesser advantage.

One possible techniques that could reduce the need for as much intervention would be to re-perform the third step of the IC algorithm after each arc is oriented due to intervention. The purpose of doing this would be to minimise the reliance on using interventions to resolve uncertain arcs. Since the third step of the IC algorithm only orients arcs based on the orientation of other arcs in the graph (for example, to prevent cycles from forming), this step may be able to make use of a newly-oriented arc to determine the orientation of other arcs in the graph.

Another factor that may reduce the need for intervention would be to have some sort of metric for deciding which variable in a pair to intervene upon

first. The program that was implemented effectively randomly chose the first variable to intervene upon in cases of completely undirected arcs ( $A - B$ ), and arcs that have an arrowhead at both ends ( $A \leftrightarrow B$ ). However, it may be possible to utilise prior probability to select the variable more likely to be a root cause (parent), which mean that only one variable would need to be intervened upon. This is because if a child variable  $A$  is chosen, the structure will be resolved as  $A \leftarrow O B$ , and variable  $B$  will need to be intervened upon as well. Alternatively, some study could be done of the rest of the causal model, in an attempt to find the most likely causal structure of the correlation and select the variable most likely to be the parent – again, meaning that only one variable would need to be intervened upon. It is also worth noting that if a common-cause latent variable is present, then both variables will need to be intervened upon regardless (unless the arc is already partially directed), so in such cases it does not particularly matter which variable is intervened upon first.

And of course, the techniques used in the IC-Intervention algorithm could be applied only partially – intervention could occur only on certain variables, which may still give additional information over what would otherwise be possible (as mentioned in section 5.2). Alternatively, the algorithm could be modified to only examine arcs of interest (as opposed to resolving every partially-directed arc), since it is unlikely that every arc within the network will be of significant interest to a particular piece of research.

## 7 Conclusions and Future Work

Causal analysis has come a long way in the past twenty years, and now there is a range of techniques available for causal discovery. However, constraint-based learning algorithms suffer from a significant drawback, and that is that they are often unable to recover the exact causal model. They frequently leave arcs undirected, frequently meaning that useful conclusions cannot be drawn. Further, even when the algorithm is able to partially direct an arc, it is rarely able to recover information about any latent variables that may be affecting the model. However, through the use of intervention variables, the IC-Intervention algorithm is able to enhance existing learning techniques to the point where they are able to (if the entire model is intervenable) recover all direct causal influences, detect all simple latent variable correlations, and detect the potential presence of more complex latent structures, if not any details about them. However, there is still much room for future research in the area.

As mentioned earlier, there are a number of implementational issues with the experimental test program, which ideally would be fixed before this algorithm would be used for any practical. First of these would be to implement the full FCI algorithm in the program, as opposed to the IC algorithm. Further, it would be good to incorporate the idea mentioned in section 6.3, where step 3 of the IC algorithm could be repeated after each intervention, in case the newly oriented arc could possibly shed any light on the structure of the rest of the network. While neither of these issues poses any significant challenge, due to time constraints we were unable to complete these tasks, particularly given that they only affect the efficiency of the program, not its end result. Other ideas discussed in section 6.3, such as an analysis of which variable to intervene upon first, would make for interesting research in the future. Of course, while the

experimental results presented here are promising, in order to be considered successful this algorithm would have to be testing using real statistical sample data. While the use of an independence oracle can accurately represent the basic functionality of the algorithm, the use of a statistical correlation significance test would confirm the algorithm's potential for practical application.

Another potential avenue that could be explored in this area is the incorporation of latent variables into metric-based causal learners. This project has focused on constraint-based learners, as they have been better explored historically, but recent developments in the field of metric-based learners show great promise for them. Additionally, the incorporation of latent variables into metric-based learners would likely not require the use of intervention, which is one of the major practical hindrances to the IC-Intervention algorithm. However, investigation of this alternative possibility is again beyond the scope of this project.

The results presented in this paper appear very positive. Given that the main problems with constraint-based algorithms are their inability to provide specific, fully-directed graphs, and their inability to discover significant information about latent variables, the use of the IC-Intervention algorithm appears to hold much potential for the future of causal discovery techniques. Even if the use of intervention is not entirely practical for a particular situation, the techniques described here should highlight for researchers the variables where intervention would prove most useful. The details of how much extra information can be recovered with partial information is a question for another study. For situations where researchers are able to intervene fully upon the variable space, though, this algorithm appears able to recover the entire direct causal structure, as well as significant information about the latent structure of a causal model.

## References

- [1] Sherif Z. Abdel-Rahman, Randa A. El-Zein, Joseph B. Zwischenberger, and William W. Au. Association of the NAT1\*10 genotype with increased chromosome aberrations and higher lung cancer risk in cigarette smokers. *Mutation Research*, 398(1-2):43–54, February 1998.
- [2] Herbert B. Asher. *Causal Modeling*. Number 003 in Quantitative Applications in the Social Sciences. Sage University Papers, Beverly Hills, London, first edition, 1976.
- [3] Ingo A. Beinlich, Henri J. Suermondt, R. Martin Chavez, and Gregory F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Proc. Second European Conf. on Artificial Intelligence in Medicine*, pages 247–256, 1989.
- [4] J. Bunker, W. Forrest, F. Mosteller, and L. Vandam. The national halothane study. Technical report, National Research Council, 1969.
- [5] David M. Chickering. A transformational characterization of equivalent Bayesian network structures. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87–98, San Mateo, CA, USA, 1995. Morgan Kaufmann.

- [6] Gregory F. Cooper and Changwon Yoo. Casual discovery from a mixture of experimental and observational data. In Kathryn Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 116–125, San Mateo, CA, USA, 1999. Morgan Kaufmann.
- [7] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [8] Allen L. Edwards. *An Introduction to Linear Regression and Correlation*. W. H. Freeman and Company, San Fransisco, CA, USA, first edition, 1976.
- [9] Gal Elidan and Nir Friedman. The information bottleneck EM algorithm. In Christopher Meek and Uffe Kjaerulff, editors, *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 200–208, San Mateo, CA, USA, 2003. Morgan Kaufmann.
- [10] Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In Douglas H. Fisher, editor, *Proc. 14th International Conference on Machine Learning*, pages 125–133, San Francisco, CA, USA, 1997. Morgan Kaufmann.
- [11] Nir Friedman. The Bayesian structural EM algorithm. In Gregory Cooper and Serafin Moral, editors, *Proceedings of the fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 129–138, San Francisco, CA, USE, 1998. Morgan Kaufmann.
- [12] Clark N. Glymour, Richard Scheines, Peter Spirtes, and Kevin Kelly. *Discovering Causal Structure*. Orlando Academic Press, Orlando, FL, USA, 1987.
- [13] Jiawei Han and Micheline Kamber. *Data Mining*. Morgan Kaufmann, 2001.
- [14] Germund Hesslow. Two notes on the probabilistic approach to causality. *Philosophy of Science*, 43(2):290–292, September 1976.
- [15] Kevin B. Korb, Lucas R. Hope, Ann E. Nicholson, and Karl Axnick. *Varieties of Causal Intervention*, volume 3157 of *Lecture Notes in Computer Science*, pages 322–331. Springer Berlin / Heidelberg, 2004.
- [16] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, London, UK, first edition, 2004.
- [17] Kevin B. Korb and Erik Nyberg. The power of intervention. *Minds and Machines*, 2006.
- [18] Wai Lam and Faheim Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(4), 1994.
- [19] MDMC WebGroup. MDMC software - CaMML (homepage). website, March 2006. <http://www.datamining.monash.edu.au/software/camml/>.

- [20] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, School of Computer Science, University of California, Berkeley, 2002.
- [21] Judea Pearl. *Causality : models, reasoning, and inference*. Cambridge University Press, 2000.
- [22] John O Rawlings. *Applied Regression Analysis: A Research Tool*. Wadsworth, 1988.
- [23] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
- [24] Richard Scheines. Tetrad project homepage. website, April 2006. <http://www.phil.cmu.edu/projects/tetrad/tetrad2.html>.
- [25] Peter Spirtes, Clark N. Glymour, and Richard Scheines. *Causation, Prediction and Search*. Number 81 in Lecture Notes in Statistics. Springer-Verlag, New York, NY, USA, first edition, 1993.
- [26] Tom S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In Piero Bonissone, Max Henrion, Laveen Kanal, and John Lemmer, editors, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 255–268, New York, NY, USA, 1991. Elsevier Science.
- [27] Chris S. Wallace, Kevin B. Korb, and Honghua Dai. Causal discovery via MML. In *International Conference on Machine Learning*, pages 516–524, 1996.
- [28] Sewall Wright. The method of path coefficients. *The Annals of Mathematical Sciences*, 5(3):161–215, September 1934.
- [29] Mary S. Younger. *Handbook for linear regression*. North Scituate, Mass., 1979.

## A Tested Networks

The networks that the algorithm was tested on, as well as the network that was recovered both with the standard IC algorithm, and with the IC-Intervention algorithm, are listed in Table 4. In all of the original networks, any variable with its name in brackets was marked as being a latent variable. In recovered networks, an arc with an end of type  $\rightarrow$  may or may not have an arrowhead at that end; also the presence of a bi-directional arc indicates the presence of a latent variable.

These networks were chosen to present the following test cases:

- a) Simple network with no latent variables (modified version of the metastatic cancer network from [16])
- b) A single latent variable
- c) Complex latent structure affecting a single correlation
- d) A slightly larger, more complex network, with multiple latent variables (reproduced from [25])

Network	Original network	IC recovered	IC-Intervention recovered
a)			
b)			
c)			
d)			
e)			

Tab. 4: Experimentally tested networks and associated recovered networks

- e) A modified version of above, with slightly more complex latent structures. Also illustrates a latent variable and a direct influence affecting the same correlation.

## **B Source Code**

The source code for the experimental software is included on the following pages.