A BAYESIAN LOCAL CAUSAL DISCOVERY FRAMEWORK

by

Subramani Mani

MBBS, Medical College Trivandrum, 1987

MS (Computer Science), University of South Carolina, 1994

Submitted to the Graduate Faculty of

the School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2005

UNIVERSITY OF PITTSBURGH

SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Subramani Mani

It was defended on

May 17, 2005

and approved by

Gregory F. Cooper, Department of Medicine and the Intelligent Systems Studies Program, University of Pittsburgh

Michael M. Wagner, Department of Medicine and the Intelligent Systems Studies Program, University of Pittsburgh

Peter Spirtes, Department of Philosophy, Carnegie Mellon University

Bruce G. Buchanan, University Professor of Computer Science (Emeritus), Department of Computer Science, University of Pittsburgh

Dissertation Director: Gregory F. Cooper, Department of Medicine and the Intelligent Systems Studies Program, University of Pittsburgh

A BAYESIAN LOCAL CAUSAL DISCOVERY FRAMEWORK

Subramani Mani, PhD

University of Pittsburgh, 2005

This work introduces the Bayesian local causal discovery framework, a method for discovering unconfounded causal relationships from observational data. It addresses the hypothesis that causal discovery using local search methods will outperform causal discovery algorithms that employ global search in the context of large datasets and limited computational resources. Several Bayesian local causal discovery (BLCD) algorithms are described and results presented comparing them with two well-known global causal discovery algorithms PC and FCI, and a global Bayesian network learning algorithm, the optimal reinsertion (OR) algorithm which was post-processed to identify relationships that under assumptions are causal.

Methodologically, this research formalizes the task of causal discovery from observational data using a Bayesian approach and local search. It specifically investigates the so called Y structure in causal discovery and classifies the various types of Y structures present in the data generating networks. It identifies the Y structures in the Alarm, Hailfinder, Barley, Pathfinder and Munin networks and categorizes them. A proof of the convergence of the BLCD algorithm based on the identification of Y structures, is also provided. Principled methods of combining global and local causal discovery algorithms to improve upon the performance of the individual algorithms are discussed. In particular, a post-processing method for identifying plausible causal relationships from the output of global Bayesian network learning algorithms is described, thereby extending them to be causal discovery algorithms.

In an experimental evaluation, simulated data from synthetic causal Bayesian networks representing five different domains, as well as a real-world medical dataset, were used. Causal

discovery performance was measured using precision and recall. Sometimes the local methods performed better than the global methods, and sometimes they did not (both in terms of precision/recall and in terms of computation time). When all the datasets were considered in aggregate, the local methods (BLCD and BLCDpk) had higher precision. The general performance of the BLCD class of algorithms was comparable to the global search algorithms, implying that the local search algorithms will have good performance on very large datasets when the global methods fail to scale up. The limitations of this research and directions for future research are also discussed.

Keywords: Causality, Causal Bayesian networks, Causal discovery, Global search, Local search, Markov blanket, BLCD, Y structure, Infant mortality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

We are always searching for causes and effects. My hope is that this research will shed light and provide guidance for this task. I dedicate this dissertation to all my teachers from grade school to grad school who opened the doors and windows of inquiry for me.

First and foremost, I would like to express my gratitude to my advisor professor Greg Cooper for his guidance through the twists and turns of this research. I also sincerely thank my other committee members professor Peter Spirtes, professor Bruce Buchanan and professor Mike Wagner for helpful discussions and critical feedback. I express my gratitude to Dr. Christoph Lehmann and Dr. Michael Neufeld for grading the output of the algorithms obtained from the Infant dataset. I would also like to convey my special thanks to Dr. Joseph Ramsey for help with the latest version of the Tetrad program.

I recall with great appreciation the interesting discussions with my friends and fellow students, in particular, Changwon Yoo and Mehmet Kayaalp that provided an enthusiastic backdrop to pursue my research.

I take this opportunity to thank the National Library of Medicine and the Mellon foundation for fellowship support. Finally, my heartfelt thanks go to my family for making this possible.

December, 2005 Subramani Mani

1.0 WHY CAUSAL DISCOVERY?

1.1 INTRODUCTION

Seeking causes for various phenomena is a significant part of human endeavor. It has been pointed out that "Causality is explanation" (Salmon, 1997), and explanation contributes to *understanding*. Consider the phenomenon of ozone layer depletion. Scientists studying the ozone layer need to measure the magnitude of loss over time, generate causal postulates and verify them. An ideal approach will involve not only identification of mechanisms responsible for the effect, but also intervention so as to arrest and maybe reverse the trend. Causality is the key to this understanding. Causal knowledge aids planning and decision making in almost all fields. For example, in the domain of medicine, determining the cause of a disease helps in prevention and treatment.

To make the world a better place to live, causal knowledge is the key. Causal knowledge has the potential to tell us the effects of manipulation of the world. It provides explanation for observed phenomena based on past interventions and assessment of outcomes. Causal knowledge also gives us the insight to understand the concurrent causal mechanisms acting in a domain enabling us to plan manipulations or interventions with desirable outcomes. In other words, we will be able to predict the effects of changing (manipulating) nature based on causal knowledge.

As a modern example, discovering causal influences is of paramount importance in systems biology. By systems biology, we refer to the paradigm shift currently occurring in biology. Instead of just studying single genes and proteins, researchers are investigating whole genomes and proteomes. The advances being made in genomics, proteomics, and metabolic pathways research are contributing to this change. The completion of the human genome

project, the development of the various types of microarray platforms for mRNA expression patterns, protein chips, and other technologies for studying biomolecules are generating a deluge of biological data. This calls for newer methods and algorithms to understand the data. Traditional ways of concentrating on one gene or protein may not be enough. A global perspective has to be developed to obtain the "big picture" and this requires addressing both technical and conceptual issues (Lander, 1999). To accomplish the goals of systems Biology we have to understand the complex causal interactions in the genome and the proteome.

Well designed experimental studies, such as randomized controlled trials, are typically employed in assessing causal relationships. Classically, the value of the variable postulated to be *causal* is set randomly and its effects are measured. These studies are appropriate in certain situations, for example, animal studies and studies involving human subjects that have undergone a thorough procedural and ethical review.

Based on meta analysis of randomized (experimental) and nonrandomized (observational) studies in healthcare, researchers have found marked correlation between the observational and experimental studies (Benson & Hartz, 2000; Ioannidis et al., 2001). Benson and Hartz focused on clinical studies conducted between 1985 and 1998 and identified more than one hundred published reports related to 19 different treatments. They found that only two of the nineteen treatments had a difference in outcome that was statistically significant between experimental and observational studies (Benson & Hartz, 2000). Ioannidis and others did a larger survey looking at published studies between 1966 and 2000 covering 45 diverse topics based on 240 clinical trials. They found a good correlation between randomized and nonrandomized studies. However, they also noted differences in seven areas that could not be explained by chance (Ioannidis et al., 2001).

Though it is not clear if these findings related to healthcare are generalizable to other areas of inquiry, the potential for observational data as a valid source for discovery science is reinforced by these studies. Observational data is passively and non-invasively collected in routine settings. There is no controlled experimental manipulation of domain variables for data collection. Census data, vital statistics, most business and economic data, astronomical data (e.g., satellite imagery data) and healthcare data routinely collected are some common examples of observational data. Since experimental studies involve deliberate manipulation

of variables and subsequent observation of the effects, they have to be designed and executed with care and caution. Experimental studies may not be feasible in many contexts due to ethical, logistical, or monetary cost considerations. These practical limitations of experimental studies heighten the importance of exploring, evaluating and refining techniques to learn more about causal relationships from observational data. The goal is not to replace experimental studies, which are extremely valuable in science, but rather to augment, refine and guide experimental studies when feasible. If pointers to interesting causal relationships could be obtained from observational studies, those causal influences could be more rigorously tested and evaluated in experimental settings more efficiently. In those areas of inquiry where experimental studies are not possible or feasible, then causal insight may need to rely primarily on observational data.

Moreover, we need discovery methods and efficient algorithms that will scale up to handle the enormous amounts of data generated continuously in diverse domains. The efficiency and scalability requirements lead us to the following hypothesis for causal hypothesis.

1.2 HYPOTHESIS FOR CAUSAL DISCOVERY

Before stating our hypothesis formally we introduce some definitions and provide a context $\mathcal{C}$ for causal data mining.

Large datasets Datasets from which it is not feasible to learn a global causal model using *limited computational resources* (defined below). The following datasets would be considered as large datasets for our study purposes.

1. Datasets with more than hundred variables or more than ten thousand records. Examples include vital statistics records, multi-center studies, and clinical patient records in medical centers.
2. Datasets with more than one thousand variables, such as gene expression datasets.

Anytime framework A context in which an *anytime algorithm* would be useful. An anytime algorithm has the property of progressively improving its results over time. These

algorithms are usually controlled by a meta-level decision procedure to evaluate the output and to stop or continue the computation (Russell & Norvig, 1995, page 844).

Local search (LS) In this dissertation research the focus of a local search methodology is on discovering submodels (subgraphs) of the causal Bayesian network generating the data (for example, pair-wise causal relationships, a node and its parents (direct causes), a node and its children (direct effects)), rather than attempting to build a global causal model. The LS considers a small subset of the total number of variables—for example, triplets or quads of variables at one time. The LS also incorporates other background knowledge in the form of priors. The priors could be special ("W") variables, a temporal ordering of the variables in the dataset, or already known pair-wise causal relationships.

Limited computational resources Limited processor speed, main memory and running time. A typical current example would be a PC with a 3 GHz processor speed, 1 GB RAM, and a run time of two weeks.

Validity In this dissertation research validity of an output is the *degree* to which a relationship is causally correct in some context. For causal discovery algorithms, validity of the output can be assigned based on how it compares to a reference standard. Validity can be studied by categorizing it into three different types—content, criterion-related and construct.

The *content validity* of an algorithm can be ascertained by the underlying theory, assumptions and the correctness of the algorithm. If the underlying theory is sound, the assumptions are correct, and there is a proof of correctness for the algorithm, content validity can be assigned. This is the most basic concept of validity and is also known as "face" validity.

Criterion-related validity can be assessed by comparing the output of the algorithm to a reference gold standard.

Construct validity is similar to criterion-related validity but needs a more sophisticated and rigorous approach (Friedman & Wyatt, 1997). For example, evaluation of a potential "causal" output can be confirmed by a randomized controlled trial.

In general, validity can be defined subjectively by the expert user based on his knowledge of the domain. The validity scale could be qualitative or quantitative.

Better Performance For grading the performance of a causal discovery algorithm the focus will be on causal structure. The performance will be primarily assessed using simulated data from expert designed causal networks as the true structure is known and hence the performance of the algorithm can be assessed as a function of the true positives, false positives, true negatives, and false negatives.

In particular the following two metrics will be computed for the output of the algorithm:

$$Precision = \frac{TPO}{TO} \tag{1.1}$$

$$Recall = \frac{TPO}{TP} \tag{1.2}$$

where TPO is the *Total number of true positive relationships output*, TO is the *Total number of relationships output*, and TP is the *Total number of true positive relationships in the network*.

Better performance is defined as higher precision and higher recall.

The context $\mathcal{C}$ for our causal data mining hypothesis incorporates the following features:

1. Large datasets.
2. An *anytime* framework.
3. Limited computational resources.

Hypothesis: *Causal discovery using local search methods in context $\mathcal{C}$ will have better performance compared to the causal discovery methods using global search described in Chapter 3.*

2.0 BACKGROUND: FRAMEWORK FOR CAUSAL DISCOVERY

Information technology (IT) is at the forefront of increasing data generation and storage which has resulted in the availability of larger and larger datasets in various domains. It is important to have efficient and *anytime* approaches to discover meaningful patterns in these large volumes of data using available computational resources. This dissertation addresses this issue from the perspective of discovering causal relationships.

This section is organized as follows. We first provide a brief introduction to Bayesian networks (BN) in Section 2.1 and causal Bayesian networks (CBN) in Section 2.2. We define and discuss "causal influence" in Section 2.3, and introduce the basic assumptions used for causal discovery in Section 2.4. Bayesian scoring of causal Bayesian networks is discussed in Section 2.5. *Bayesian model averaging* takes model uncertainty into account rather than assume that a single model is correct. It is described in Section 2.6. Section 2.7 discusses selective Bayesian model averaging and model selection for causal discovery. Section 2.8 describes the method we use for handling missing data in practice.

2.1 BAYESIAN NETWORKS

Our framework for causal discovery is founded on Bayesian networks. BNs are directed acyclic graphs (DAGs) with the vertices (nodes[1]) representing observed variables in a domain and the directed edges denoting dependence relationships between the variables. The probabilistic relationships among the variables represented in a BN are quantified by marginal probabilities (for root nodes) and conditional probabilities (for non-root nodes). The joint

[1]We use vertices and nodes interchangeably.

probability distribution of the variables represented in a domain can be expressed compactly using a BN and can be factorized as follows:

$$\mathrm{P}(X_1, X_2, \ldots, X_{n-1}, X_n) = \prod_{i=1}^{n} \mathrm{P}(X_i \,|\, \mathrm{Pa}(X_i)) \tag{2.1}$$

where:

- $X_1, X_2, \ldots, X_{n-1}, X_n$ are the vertices of the BN.
- $\mathrm{Pa}(Xi)$ denotes the set of parents of the node X_i in the BN (X_j is a parent of X_i iff there is a directed edge from X_j to X_i).

See (Neapolitan, 1990; Pearl, 1991; Heckerman, 1996) for more details.

Our focus is on learning causal Bayesian networks (see Section 2.2) from data and the natural question is why should we try to learn these models from data. Initially Bayesian networks were built as knowledge-based systems. The structure and the parameters were specified by experts (Beinlich et al., 1990; Andreassen et al., 1987; Heckerman et al., 1992). It was very labor intensive and challenging for experts to specify precisely the prior and conditional probabilities that parameterized the models. For various domains, experts had problems assessing a full-fledged causal structure, or the parameters, and in some situations both. Hence researchers started to focus on data—initially for parameter estimation, but subsequently to learn both the network structure as well as the probabilities associated with it.

2.2 CAUSAL BAYESIAN NETWORKS

A causal Bayesian network (or *causal network* for short) is a Bayesian network in which each arc is interpreted as a direct causal influence between a parent node (variable) and a child node, relative to the other nodes in the network (Pearl, 1991). For example, if there is a directed edge from A to B ($A \longrightarrow B$), node A is said to exert a causal influence on node B. Figure 1 illustrates the structure of a hypothetical causal Bayesian network structure, which

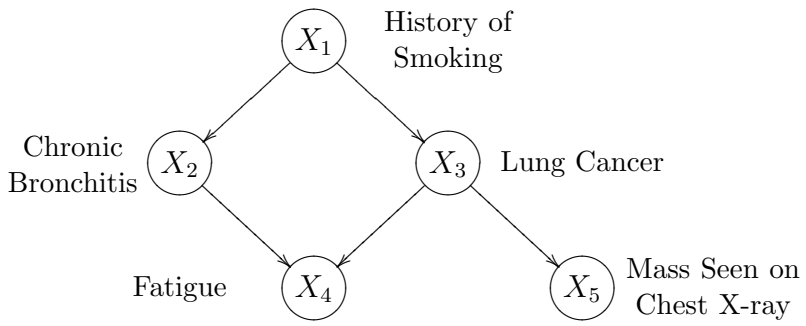


Figure 1: A hypothetical causal Bayesian network structure

contains five nodes. The states of the nodes and the probabilities that are associated with this structure are not shown.

The causal network structure in Figure 1 indicates, for example, that a *History of Smoking* can causally influence whether *Lung Cancer* is present, which in turn can causally influence whether a patient experiences *Fatigue* or presents with a *Mass Seen on Chest X-ray*.

The *independence map* or I-map of a causal network is the set of dependence/independence relationships between individual variables or sets of variables unconditioned or conditioned on other variables or sets of variables. The I-map of the causal network $W_1 \longrightarrow X \longleftarrow W_2$ is as follows:

- $W_1 \perp\!\!\!\perp W_2$
- $W_1 \not\perp\!\!\!\perp X$
- $W_1 \not\perp\!\!\!\perp X | W_2$
- $W_1 \not\perp\!\!\!\perp X$
- $W_2 \not\perp\!\!\!\perp X | W_1$
- $W_1 \not\perp\!\!\!\perp W_2 | X$

2.3 CAUSAL INFLUENCE, CONFOUNDED AND UNCONFOUNDED CAUSAL RELATIONSHIPS

We define the *causal influence* of a variable X on variable Y using the *manipulation criterion* (Spirtes et al., 1993; Glymour & Cooper, 1999). The manipulation criterion states that if we had a way of setting just the values of X and then measuring Y, the causal influence of X on Y will be reflected as a change in the conditional distribution of Y. That is, there exists values x_1 and x_2 of X such that $P(Y|$ set $X = x_1) \neq P(Y|$ set $X = x_2)$. We now introduce the types of causal influences (relationships) encountered in our framework.

In a causal Bayesian network an arc between any pair of nodes represents a causal influence. These causal relationships can be termed as confounded or unconfounded based on the following criteria. The arcs and node pairs can be categorized using the following framework. Each pair (X, Y) is categorized as follows:

Causal and unconfounded pair (CUP) If CUP(X,Y), then there is a directed path from X to Y, and there is no common ancestor W that has a directed path to X and a directed path to Y that does not traverse X. A directed path from node X to node Y is a set of one or more directed edges originating from X and ending in Y. The nodes *Lung Cancer* and *Mass seen on Chest X-ray* in Figure 1 are *causal and unconfounded*, that is CUP(X_3,X_5) holds.

Causal and confounded pair (CCP) If CCP(X,Y) holds then there is a directed path from X to Y, and there is a common ancestor W that has a directed path to X, and a directed path to Y that does not traverse X. The nodes *Chronic Bronchitis* and *Fatigue* in Figure 1 are *causal and confounded* (*History of Smoking* is a common ancestor).

For completeness we now introduce the two other types of relationships encountered between two variables X and Y in a causal Bayesian network.

Confounded-only pair (COP) There is no directed path between X and Y, and there is a common ancestor W that has a directed path to X, and a directed path to Y that does not traverse X. The nodes *Chronic Bronchitis* and *Lung Cancer* in Figure 1 have the *confounded-only pair* relationship.

Independent pair (IP) There is no d-connecting path (Pearl, 1991) between X and Y. See Section 3.1.1 for an explanation of d-separation and d-connectivity.

The arcs of a CBN can be categorized as given below:

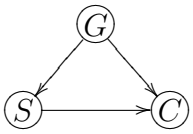
Causal and unconfounded arc (CUA) If CUA(X,Y), then there is an arc from X to Y, and there is no common ancestor W that has a directed path to X and a directed path to Y that does not traverse X. The arc between *Lung Cancer* and *Mass seen on Chest X-ray* in Figure 1 is *causal and unconfounded*, that is CUA(X_3,X_5) holds.

Causal and confounded arc (CCA) If CCA(X,Y), then there is an arc from X to Y, and there is a common ancestor W that has a directed path to X and a directed path to Y that does not traverse X. The arc between *Chronic Bronchitis* and *Fatigue* in Figure 1 is *causal and confounded* (*History of Smoking* is a common ancestor).

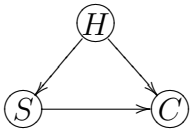
Note that all causal and unconfounded arcs (CUA) are causal and unconfounded pairs (CUP) but not vice versa. Likewise, all causal and confounded arcs (CCA) are causal and confounded pairs (CCP) but not vice versa. Note also that there cannot exist confounded-only or independent arcs.

In causal discovery, we are usually interested in identifying both confounded (usually by a measured variable) and unconfounded causal relationships. Consider the three hypothetical models in Figure 2. Assume that G stands for a Gene, S for Smoking, and C for Cancer, and these variables have two states—present and absent. Note that H stands for a hidden variable. Model (1) has the measured confounder G, Model (2) has a hidden (unmeasured) confounder H, and Model (3) is unconfounded. Model (1) and model (3) are informative. For example, if G is causing a section of the population to smoke and also causing lung cancer (C), an effective intervention strategy could be to focus on the segment of the population without G and persuade them to stop smoking. Likewise, advocating cessation of smoking is a good interventional strategy to reduce the incidence of lung cancer based on model (3). The causal effect of S on C can be assessed from observational data. Model (2) is a lot less informative. Because of the hidden confounder H, the causal effect of S on C cannot be quantified with confidence from observations (not interventions) of S and C only.

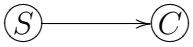
Broadly speaking causal discovery can also encompass *acausal discovery*, that is, rela-



(1)



(2)



(3)

Figure 2: Three hypothetical causal models in which S causes C. S and C are confounded by a measured variable represented by G in Model (1), and a hidden confounder H in Model (2). There is no confounding variable (measured or hidden) in Model (3).

tionships of the form "X does not causally influence Y". However, for purposes of this dissertation we concentrate on causal discovery and do not focus on acausal discovery.

2.4 ASSUMPTIONS FOR CAUSAL DISCOVERY

In this section we describe the basic assumptions of our causal discovery framework. Additional assumptions that some causal discovery algorithms may require will be introduced with the respective algorithm descriptions.

2.4.1 The causal Markov condition

The **causal Markov condition** (CMC) gives the independence relationships[2] that are specified by a causal Bayesian network:

> A variable is independent of its non-descendants (i.e., non-effects) given just its parents (i.e., its direct causes).

According to the causal Markov condition, the causal network in Figure 1 is representing that the chance of a *Mass Seen on Chest X-ray* will be independent of a *History of Smoking*, given that we know whether *Lung Cancer* is present or not.

The CMC is representing the "locality" of causality. This implies that indirect (distant) causes become irrelevant when the direct (near) causes are known. For example, if we know the status of *Lung Cancer* (present/absent), knowledge of *History of Smoking* does not give any additional information to enhance our understanding of the effect variable (*Mass seen on chest X-ray*). The CMC can fail in certain situations as illustrated by the following two examples adapted from (Cooper, 1999).

Consider the hypothetical CBN in Figure 3. If we sum over and marginalize out the X_2, we get the CBN shown in Figure 4. This demonstrates that the absence of a statistical dependency between two variables X and Y does not necessarily mean that there is no causal relationship between them when hidden variables are also considered.

High Blood Cholesterol : no, moderate, severe

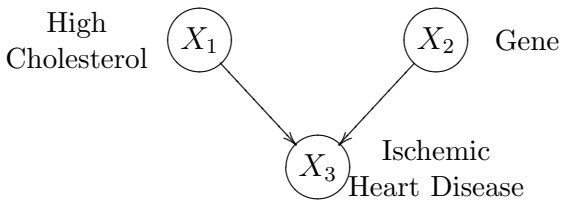
Coronary Artery Disease : absent, moderate, severe

Myocardial Infarction : absent, present

Figure 5 shows a CBN that depicts the following hypothetical causal relationships: *High Blood Cholesterol* causally influences *Coronary Artery Disease* which in turn causally influences *Myocardial Infarction*. Each of these variables take one of the corresponding values given above.

Assume that the CMC correctly implies that X_1 and X_3 are independent given X_2. Now consider the variables and their states where *Coronary Artery Disease* takes values "absent"

[2]We use the terms *independence* and *dependence* in this section in the standard probabilistic sense.



$$
\begin{aligned}
\mathrm{P}(X_1\text{=present}) &= 0.5 \\
\mathrm{P}(X_2\text{=present}) &= 0.5 \\
\mathrm{P}(X_3\text{=present}) \mid X_1\text{=present}, X_2\text{=present}) &= 0.1 \\
\mathrm{P}(X_3\text{=present}) \mid X_1\text{=present}, X_2\text{=absent}) &= 0.9 \\
\mathrm{P}(X_3\text{=present}) \mid X_1\text{=absent}, X_2\text{=present}) &= 0.9 \\
\mathrm{P}(X_3\text{=present}) \mid X_1\text{=absent}, X_2\text{=absent}) &= 0.1
\end{aligned}
$$

Figure 3: A CBN with three nodes X_1, X_2, and X_3. [Modified from (Cooper, 1999)]

and "present" instead of the three level given earlier. If we now condition on the value *Coronary Artery Disease* = present, it is possible that X_1 may not be independent of X_3. A "severe" *High Blood Cholesterol* causes "severe" *Coronary Artery Disease* that in turn causes myocardial infarction. In this scenario, knowledge that "*Coronary Artery Disease* = present" does not render *High Blood Cholesterol* and *Myocardial Infarction* independent (see Figure 6).

This example shows that the independence properties of a CBN implied by the CMC may vary if the number of states of one or more variables in the CBN is modified (Cooper, 1999).

Another point worth noting is that based on quantum theory causality is not local (Herbert, 1985; Cooper, 1999) and this may cause CMC to fail. However, we note that as with Einsteinian physics not negating Newton's laws in the macroscopic world, local conditioning given by CMC is not generally negated by quantum mechanics in the macro world.

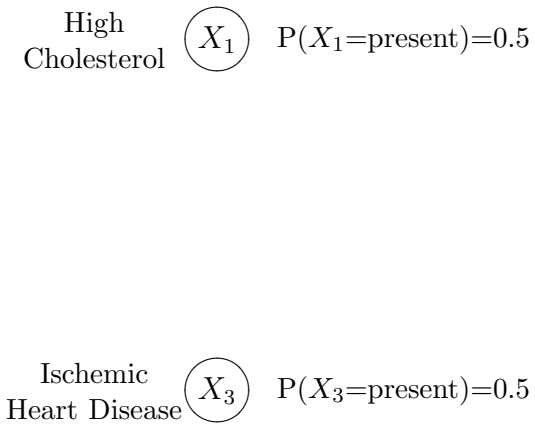


Figure 4: A CBN with two nodes X_1, and X_3. [Modified from (Cooper, 1999)]

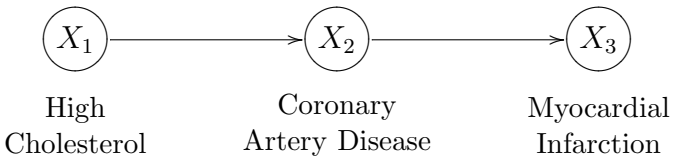
2.4.2 The causal faithfulness condition

While the causal Markov condition specifies independence relationships among variables, the **causal faithfulness condition** (CFC) specifies *dependence* relationships:

> Variables are dependent unless their independence is implied by the causal Markov condition.

For the causal network structure in Figure 1, three examples of the causal faithfulness condition are (1) *History of Smoking* and *Lung Cancer* are probabilistically dependent, (2) *History of Smoking* and *Mass Seen on Chest X-ray* are dependent, and (3) *Mass Seen on Chest X-ray* and *Fatigue* are dependent. The intuition behind that last example is as follows: the existence of a *Mass Seen on Chest X-ray* increases the chance of *Lung Cancer* which in turn increases the chance of *Fatigue*; thus, the variables *Mass Seen on Chest X-ray* and *Fatigue* are expected to be probabilistically dependent. In other words, the two variables are dependent because of a common cause (i.e., a confounder).

The CFC is related to the notion that causal events are typically correlated in observational data. The CFC relates causal structure to probabilistic dependence. The CFC generally holds in most situations but it can also fail. The following discussion and examples are based on (Cooper, 1999).



High Blood Cholesterol : no, moderate, severe

Coronary Artery Disease : absent, moderate, severe

Myocardial Infarction : absent, present

Figure 5: A CBN with three nodes X_1, X_2, and X_3. [Modified from (Cooper, 1999)]

2.4.2.1 The problem of deterministic relationships Assume that all the variables in Figure 7 are binary and the probability distributions are as follow:

$\mathrm{P}(X_1{=}\text{yes}) = 1$

$\mathrm{P}(X_2{=}\text{yes} \mid X_1{=}\text{yes}) = 1$

$\mathrm{P}(X_2{=}\text{no} \mid X_1{=}\text{no}) = 1$

$\mathrm{P}(X_3{=}\text{yes} \mid X_2{=}\text{yes}) = 1$

$\mathrm{P}(X_3{=}\text{no} \mid X_2{=}\text{no}) = 1$

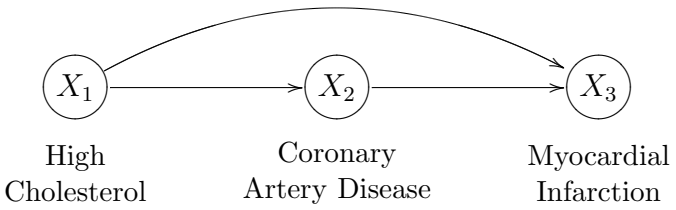


Figure 6: A CBN with three nodes X_1, X_2, and X_3. [Modified from (Cooper, 1999)]

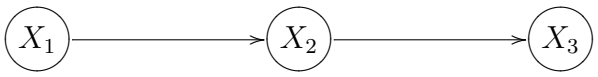


Figure 7: A CBN with three nodes X_1, X_2, and X_3.

Note that all the three variables X_1, X_2, and X_3 are related in a deterministic way. All the three variables are always in the state "yes". Since all the values are the same (yes), we cannot infer from observational data the potential effect of a manipulation.

Now assume that P(X_1=yes) = 0.5. All the conditional probabilities remain the same. Knowing the value of X_3 tells us the value of X_1 and conditioning on X_2 has no effect.

In both these situations CFC is violated. We also emphasize that a deterministic relationship is not a necessary condition for the violation (see Figure 3).

2.4.2.2 Goal oriented systems A class of systems where CFC can fail is the framework of *goal oriented systems*. Consider the clinical situation represented by Figure 8. H stands

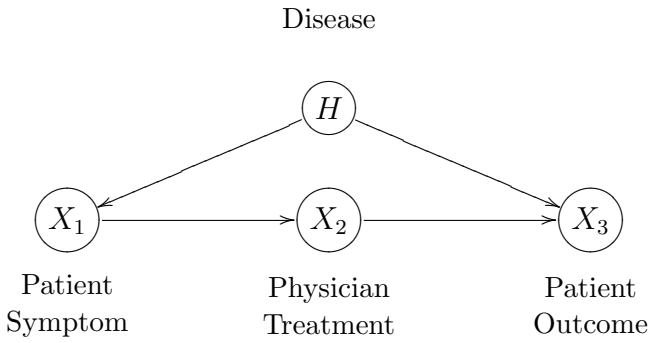


Figure 8: A CBN with three observed nodes—X_1, X_2, X_3, and one hidden node H. [Modified from (Cooper, 1999)]

for a hidden disease. Assume that it is a slow-growing tumor that turns malignant in a proportion of patients and gives rise to symptoms. The disease is fatal only when it turns malignant and left untreated. Once patients develop symptoms they consult a physician,

get treatment (say chemotherapy) and go into remission. If the outcome variable is five-year survival and all patients survive, physician action (X_2) and patient outcome (X_3) will be independent. This violates the CFC. But the CFC is likely to hold in systems that are not goal-oriented. The CFC is also likely to hold in goal-oriented systems if the goal is to keep a constant state of the system. Otherwise, the CFC is not likely to hold in goal oriented systems.

When we consider exhaustively all possible distributions the proportion that violate CFC are relatively very small (Spirtes et al., 1993; Meek, 1995). But such distributions when they exist can cause problems in discovering causal relationships using algorithms that assume CFC (Cooper, 1999).

Before we move on to the next property of a CBN, we introduce our notational convention. We represent sets of variables in bold and upper case, random variables by upper case letters italicized, and the value of a variable or sets of variables by lower case letters. When we say $\mathbf{X} = \mathbf{x}$, we mean an instantiation of all the variables in $\mathbf{X}$, while $X = x$ denotes that the variable X is assigned the value x. Graphs are denoted by calligraphic letters, such as $\mathcal{G}$ or upper case letters such as G or M.

We now describe another property of a CBN that is based on the Markov property. This property is called the *Markov Blanket* (MB). The MB of a node X in a CBN $\mathcal{G}$ is the union of the set of parents of X, the children of X, and the parents of the children of X (Pearl, 1991). Note that it is the *minimal* set of nodes when conditioned on (instantiated) that makes a node X independent of all the other nodes in the CBN. The MB is minimal and unique when there are no deterministic relationships in the CBN. Let $\mathbf{V}$ be the set of all variables in $\mathcal{G}$, $\mathbf{B}$ the MB of X, and $\mathbf{A}$ be $\mathbf{V} \setminus (\mathbf{B} \cup X)$. Conditioning on $\mathbf{B}$ renders X independent of $\mathbf{A}$. For example, in the hypothetical CBN shown in Figure 1, the MB of node X_4 is composed of X_2 and X_3; the MB of node X_5 is just X_3.

2.5 BAYESIAN SCORING OF COMPLETE CAUSAL MODELS

Bayesian methods for scoring CBNs were first developed by Cooper and Herskovits (Cooper & Herskovits, 1992). Subsequently a modified scoring metric was proposed by Heckerman et al. (Heckerman et al., 1995). Starting with a set of user-specified priors for network structures and parameters, the methods derive a posterior score using observational data and some basic modeling assumptions. Instead of user-specified priors, the methods also allow use of noninformative priors for both structures and parameters.

We now summarize the Bayesian scoring of a network structure. This discussion is based on (Cooper, 1999, pages 39–40). Let D be a dataset over a set of observed variables $\mathbf{V}$. Let B_S represent any arbitrary causal BN structure over $\mathbf{V}$ and $\mathcal{K}$ denote background knowledge that has bearing on the causal network over V. We can derive the posterior probability of a BN structure B_S as follows:

$$P(B_S|D,\mathcal{K}) = \frac{P(B_S,D|\mathcal{K})}{P(D|\mathcal{K})} = \frac{P(B_S,D|\mathcal{K})}{\sum_{B_S} P(B_S,D|\mathcal{K})} \tag{2.2}$$

Since $P(D|\mathcal{K})$ is a constant for all the causal structures, we can write equation 2.2 as:

$$P(B_S|D,\mathcal{K}) \propto P(B_S,D|\mathcal{K}) \tag{2.3}$$

The right hand side of equation 2.3 can be expressed as:

$$\begin{aligned} P(B_S,D|\mathcal{K}) &= P(B_S|\mathcal{K})P(D|B_S,\mathcal{K}) \\ &= P(B_S|\mathcal{K}) \int P(D|B_S,\theta_{B_S},\mathcal{K})P(\theta_{B_S}|B_S,\mathcal{K})d\theta_{B_S} \end{aligned} \tag{2.4}$$

where $P(B_S|\mathcal{K})$ is the prior probability of B_S given the background knowledge $\mathcal{K}$ that we possess; θ_{B_S} represent the parameters associated with the BN structure B_S; $P(D|B_S,\theta_{B_S},\mathcal{K})$ is the likelihood of data D assuming the network structure B_S, its parameterization θ_{B_S} and background knowledge $\mathcal{K}$; and $P(\theta_{B_S}|B_S,\mathcal{K})$ is the prior for the parameterization of the model given the network structure and background knowledge.

2.6 BAYESIAN MODEL AVERAGING (BMA)

If X and Y represent a pair of variables in $\mathbf{V}$, we can derive the posterior probability of the existence of a causal relationship $X \rightarrow Y$ using Bayesian model averaging over structures containing $X \rightarrow Y$:

$$P(X \rightarrow Y|D, \mathcal{K}) = \sum_{B_{S_{X\rightarrow Y}}} P(B_{S_{X\rightarrow Y}}|D, \mathcal{K}) \tag{2.5}$$

where $B_{S_{X\rightarrow Y}}$ denotes a causal BN structure with the causal relationship $X \rightarrow Y$. Combining equation 2.2 through equation 2.5, we can write:

$$P(X \rightarrow Y|D, \mathcal{K}) = \frac{\sum_{B_{S_{X\rightarrow Y}}} P(B_{S_{X\rightarrow Y}}|\mathcal{K}) \int P(D|B_{S_{X\rightarrow Y}}, \theta_{B_{S_{X\rightarrow Y}}}, \mathcal{K}) P(\theta_{B_{S_{X\rightarrow Y}}}|B_{S_{X\rightarrow Y}}, \mathcal{K}) d\theta_{B_{S_{X\rightarrow Y}}}}{\sum_{B_S} P(B_S|\mathcal{K}) \int P(D|B_S, \theta_{B_S}, \mathcal{K}) P(\theta_{B_S}|B_S, \mathcal{K}) d\theta_{B_S}} \tag{2.6}$$

The numerator sum ($\sum_{B_{S_{X\rightarrow Y}}}$) is over the BN structures containing the causal relationship $X \rightarrow Y$, and the denominator sum ($\sum_{B_S}$) is over the BN structures being modeled. Note that this derivation of posterior probability of a causal relationship is exhaustive and comprehensive. However, it often is not computationally feasible in the exact form shown.

The model B_S can be scored using a Bayesian metric such as the K2 metric (Cooper & Herskovits, 1992) or the BDe metric (Heckerman et al., 1995). Using these metrics the integrals in Equation 2.6 can be computed.

2.7 SELECTIVE BAYESIAN MODEL AVERAGING AND MODEL SELECTION

The number of causal DAGs is exponential in the number of variables in the DAG. For any non-trivial domain, scoring and summing over all these models in the framework of BMA is computationally intractable. This calls for identifying a much smaller subset of models that has the potential to be high-scoring. Cooper and Herskovits (Cooper & Herskovits, 1992) describe a greedy heuristic search method to restrict the number of models by limiting the

number of parents of each node. This greedy forward search can be used to arrive at a single highest scoring model in which case it is termed *model selection*. The search strategy can also be used to select a set of high scoring models. Another method is restricting the search to Markov equivalent class of models (Spirtes & Meek, 1995).

Since in our dissertation we learn local (composed of a subset of variables) causal models we do not limit our search to just a few of the possible CBNs over the modeled variables. Instead of restricting the scoring to Markov equivalent class of models, we score all the models that are causally distinct as discussed in Section 5.1.

2.8 HANDLING MISSING DATA

Data is said to be missing when values for observed variables are not available for a subset of instances. A variable is considered hidden when its value remains unmeasured for all the instances. See Heckerman et al. (Heckerman et al., 1999, pages 151–153) for a description of Bayesian scoring methods in the presence of missing data and hidden variables. Since an exact Bayesian approach is often intractable, approximation methods such as Gibbs sampling (Geman & Geman, 1984), Gaussian approximation, and maximum likelihood approximation with EM algorithm are often used. As the focus of this dissertation is not in developing methods for handling missing data, we now describe a practical and simple method that we propose to follow.

With simulated data the problem of missing data is normally not encountered. However, with most real-world datasets missing data is often present. The absence of an observation could be random or dependent on the actual states of the variables. For purposes of this discussion we assume that data is missing completely at random. Researchers have used different approaches to the problem of missing data. A simple approach is to discard instances with missing data. However, this could result in a substantial reduction in sample size. For the algorithms in our prior work, we excluded instances that had a missing value in any of the three attributes considered by the algorithm at a particular time. We propose to follow the same approach for a newly introduced algorithm (BLCD), as only a subset of

the variables are considered by the algorithm for its local search strategy (See Section 5.1 for details). However, the PC and FCI algorithms consider all the variables in assessing a causal relationship. Thus excluding all those instances with missing values is not a feasible approach as little or no instances would remain. We propose to take the simple approach of treating the missing value as a separate state of an attribute. For example, if a variable X has known values "Y" and "N", the missing value is assigned "U" (unknown). This can lead to missing causal relationships but not incorrectly claiming causal influences. We refer to this phenomenon as the *missing value effect*. However, this adds to the computational complexity as the number of states of each variable that has even a single missing value increases by one. This increase in the number of states of variables results in larger marginal and conditional probability tables for parameter estimation.

3.0 RELATED WORK: LEARNING CAUSAL BAYESIAN NETWORKS FROM DATA

In this chapter we review the current Bayesian network learning algorithms that are useful for discovering causal structure from data. Note that some of the algorithms reviewed do not make explicit "causal" claims. Others do so, and we describe them in greater detail.

The BN learning methods can be broadly classified into constraint-based and Bayesian. The networks have a structural component and a parametric component that represents marginal and conditional probabilities of the variables. Once the structure is ascertained, it is often straightforward to estimate the parameters from data particularly in the absence of hidden confounders. Hence our focus here will be on search methods for learning the causal structure from data.

Constraint-based methods use tests of independence/dependence between two variables (X, Y) or two sets of variables (**X, Y**) given another set of variables (**Z**) (including the empty set) to add/remove edges between variables and orient them. They typically start with a complete network and delete edges based on the tests, or start with a fully disconnected network and add edges, or do both. The number of possible tests to be done in this framework is exponential in the number of variables **V** in the worst case.

Score-based methods compute the probability of the data **D** given a structure. Exhaustive model selection involves scoring all possible network structures on a given set of variables and then picking the structure with the highest score. However, heuristic search methods are typically employed to restrict the search space as the number of potential networks are exponential in the total number of variables. Researchers have proved that it is NP-hard to identify a Bayesian network that can have up to k parents for a node where $k \geq 1$ and a score greater than some constant C using a likelihood equivalent score such as the BDe

metric (see Section 3.2.4) (Chickering, 1996).

A different class of score-based methods are based on the minimal description length (MDL) principle that has its roots in information theory. These are described in Section 3.5.

All the causal discovery algorithms discussed below make two cardinal assumptions—the *causal Markov* and the *causal faithfulness* conditions, which are described in Section 2.4. Additionally, constraint-based algorithms make use of statistical tests for elucidating dependence and independence between variables or sets of variables. Hence they also need a *statistical testing assumption*. This is discussed in Section 3.4.1.

The causal Bayesian network (CBN) learning methods can also be classified as global or local based on their search methodology and their output. If the goal is to learn a unified CBN over all the model variables, the search methodology is termed global. PC and FCI are global constraint-based algorithms (Spirtes et al., 1993). If the goal of the learning procedure is to discover causal models on subsets of the model variables (for example, pairwise causal relationships), a *local search* methodology is employed. This could be over triplets of variables (restricted local search) such as in LCD (see Section 3.4.1) or over more than three variables (extended local search) such as in BLCD (see Section 5.1).

3.1 GLOBAL CONSTRAINT-BASED ALGORITHMS

In this section we discuss the PC, FCI, anytime FCI and the GS Markov blanket algorithm.

3.1.1 PC algorithm

The PC algorithm takes as input a dataset D over a set of random variables $\mathbf{V}$, a conditional independence test, and an α level of significance threshold for the test. It then outputs an essential graph that we define below. Recall that Markov equivalence (also known as independence equivalence) is a relationship based on independence that establishes an equivalence class of directed acyclic graphs over an observed set of variables $\mathbf{V}$. These DAGs are statistically indistinguishable based on independence relationships among $\mathbf{V}$. Let U be

one equivalence class of DAGs over $\mathbf{V}$. An essential graph E of U over $\mathbf{V}$ will have directed and undirected edges such that each directed edge between a pair of nodes X and Y will be represented in *all* the DAGs in U and each undirected edge between a pair X and Y in E will be represented as either $X \rightarrow Y$ or $X \leftarrow Y$ in *all* the DAGs in U (Cooper, 1999) with both arc types represented.

We first define the terms *ancestor* and *descendent*. We then introduce the concept of a *d-separating* set and outline the steps of the algorithm.

Ancestor : In a CBN, node X is said to be an ancestor of node Y if there is a directed path from X to Y.

Descendent : In a CBN, node Y is said to be a descendent of node X is there is a directed path from X to Y.

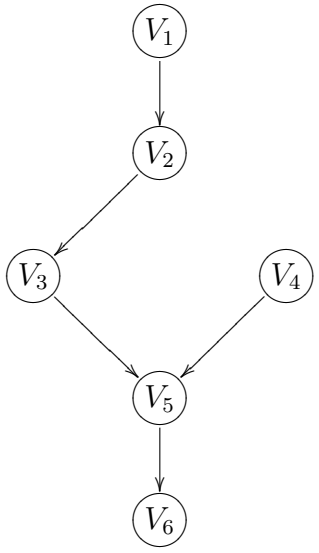


Figure 9: A hypothetical Bayesian network structure

d-separation (Pearl, 1991): Consider the DAG G in Figure 9. Assume that X and Y are vertices in G and $\mathbf{Z}$ is a set of vertices in G such that $X \notin \mathbf{Z}$ and $Y \notin \mathbf{Z}$. X and Y are said to be d-separated given $\mathbf{Z}$ iff the following property holds: there exists no *undirected path*[1] U between X and Y s.t.

[1]An undirected path between two vertices A and B in a graph G is a sequence of vertices starting with A and ending with B and for every pair of vertices X and Y in the sequence that are adjacent there is an edge between them ($X \rightarrow Y$ or $X \leftarrow Y$) (Spirtes et al., 2000, page 8)

1. every collider[2] on U has a descendant in $\mathbf{Z}$.
2. no other vertex on U is in $\mathbf{Z}$.

Likewise, if X and Y are not in $\mathbf{Z}$, then X and Y are *d-connected* given Z iff they are not d-separated given $\mathbf{Z}$.

In Figure 9 the nodes V_1 and V_6 are d-separated by V_3. The nodes V_1 and V_4 are d-connected given V_6.

The following are the salient steps of the PC algorithm (Spirtes et al., 2000, page 84–85).

1. Start with complete undirected graph composed of all the variables as vertices.
2. For each pair of variables, obtain a minimal d-separating set. The set $\mathbf{S}_{\mathrm{j}}$ is a minimal d-separating set of node X and node Y among all the d-separating sets $\mathbf{S}_{\mathrm{i}}$ if $|\mathbf{S}_{\mathrm{j}}| \leq |\mathbf{S}_{\mathrm{i}}|$ for all $\mathbf{S}_{\mathrm{i}\neq\mathrm{j}}$. For example, for the pair (X, Y), try to find the minimal conditioning set of nodes $\mathbf{S}$ (including the empty set) that satisfies $(X \perp\!\!\!\perp Y|\mathbf{S})$. This is done by starting with low order conditional tests and moving up. Note that we check only subsets of vertices currently adjacent to one endpoint or the other. If such a set $\mathbf{S}$ is found then delete edge X—Y, where X—Y represents an undirected edge between X and Y.
3. Consider each triplet of nodes X, Y, and Z where (X, Y) and (Y, Z) are adjacent[3] while (X, Z) is not. If Y is not in the d-separating set of (X, Z), the triplet is oriented as $X \rightarrow Y \leftarrow Z$.
4. Orient edges iteratively using the following rules until no more edges can be oriented.
   - If $X \rightarrow Y$ and Y—Z are present, and X and Z are not adjacent, then orient $(Y$ — $Z)$ as $(Y \rightarrow Z)$.
   - If there is a *directed path* from X to Y, and X—Y exists, then orient X — Y as $X \rightarrow Y$.
   - If $W \rightarrow X \leftarrow Y$ and X — Z, and W — Z, and Y — Z, and there is no adjacency between W and Y, then the edge X — Z can be oriented as $X \leftarrow Z$.

The PC algorithm has a worst-case time complexity that is exponential in the largest *degree* in the output graph. The degree of a vertex v refers to the number of adjacent nodes (vertices) of v. The PC is also known to be unstable in steps 2 and 3 (Spirtes et al., 1993).

[2] A node with a head to head configuration. C is a collider in $A \rightarrow C \leftarrow B$.

[3] Two nodes X and Y are said to be adjacent if there is an edge between X and Y. Initially all pairs of vertices are adjacent as the algorithm starts with a complete undirected graph.

Note that it is not guaranteed to orient all the edges as the goal is to identify the essential graph. Hence the final output can be a graph with both directed and undirected edges.

PC also makes an assumption of *causal sufficiency*. This means that all the variables of the causal network are measured and there are no latent or hidden variables. Hence PC is not designed to discover hidden variables that are common causes of any pair of observed variables.

Spirtes et al. have reported that the PC algorithm learned the Alarm network (see Section 6.3.2) from data orienting most of the edges correctly but omitted two edges of the generating graph in one trial and added an extra edge in another (Spirtes et al., 2000, page 109).

The PC algorithm was also evaluated on simulated data from artificially generated CBNs. The algorithm performed well with high sample sizes and a low average degree for the generating graph (Spirtes et al., 2000, page 116).

3.1.2 FCI algorithm

Real world data can contain hidden (unobserved or latent) variables. There are variables in a domain that were unmeasured and hence are not explicitly represented in the collected data. Another problem pertaining to real-world data is *sample selection bias*. This occurs when the values of the variables under study determine when certain instances are included in the sample (Spirtes et al., 1999). Spirtes et al. introduce the following assumption for data that is subject to sample selection bias.

3.1.2.1 Population inference assumption (Spirtes et al., 1999) The population selected by sampling criteria has the same causal structure (the statistical properties might differ due to sample selection bias) as the population about which causal inferences are to be made.

The FCI algorithm can discover causal relationships in the presence of hidden variables and selection bias. The first three steps of the FCI algorithm are similar to that of the PC algorithm. The later steps involve edge deletion and then orienting the edges using a

different set of rules compared to the PC algorithm. FCI takes as input data D representing a causal graphical structure that has the observed set of variables $\mathbf{O}$ and outputs a partial ancestral graph (PAG) of the causal structure over $\mathbf{O}$ (Spirtes et al., 1999).

We introduce some definitions that are required for understanding FCI and anytime FCI (AFCI) based on the description in (Spirtes et al., 2000) and (Spirtes et al., 1999).

1. π refers to a partial ancestral graph.
2. $\mathbf{V}$ refers to the set of all variables in a domain.
3. $\mathbf{L}$ refers to the set of **latent** variables in a domain.
4. $\mathbf{S}$ refers to the set of **selection** variables in a domain.
5. $\mathbf{O}$ refers to the set all **observed** variables.
6. $A \leftrightarrow B$ refers to a hidden cause that directly causes A and B.
7. $A \circ\!\!\rightarrow B$ refers to either $A \leftrightarrow B$ or $A \rightarrow B$.
8. $A * \!\!\rightarrow B$ refers to one of $A \leftrightarrow B$, $A \rightarrow B$, $A \circ\!\!\rightarrow B$.
9. B is a **collider** along path $< A, B, C >$ iff $A * \!\!\rightarrow B \leftarrow\!\! * C$ in π
10. An edge between B and A is **into** A iff $A \leftarrow\!\! * B$ in π.
11. An edge between B and A is **out of** A iff $A \rightarrow B$ in π.
12. $\mathbf{T}$ is $Sepset(X, Y)$ if $\mathbf{T}$ d-separates X and Y.

A DAG G with a partition of its variable set $\mathbf{V}$ into $\mathbf{O}$, $\mathbf{S}$ and $\mathbf{L}$ variables is written as $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$.

3.1.2.2 O-Equiv(Cond) (Spirtes et al., 1999) A **Cond** is a set of conditional independence relations among the variables in $\mathbf{O}$. A DAG $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$ is in **O-Equiv(Cond)** when $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$ entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} | (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$ iff $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} | \mathbf{Y}$ is in **Cond**. If $G'(\mathbf{O}, \mathbf{S}', \mathbf{L}')$ entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} | (\mathbf{Y} \cup (\mathbf{S}' = \mathbf{1}))$ iff $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$ entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} | (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$, then $G'\mathbf{O}, \mathbf{S}', \mathbf{L}')$ is in **O-Equiv**(G).

We now define a partial ancestral graph.

3.1.2.3 Partial ancestral graph (Spirtes et al., 1999) A DAG G with a partition of its variable set $\mathbf{V}$ into $\mathbf{O}$, $\mathbf{S}$ and $\mathbf{L}$ variables is written as $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$. A PAG π *represents* a DAG $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$ iff:

1. The set of variables in π is $\mathbf{O}$.
2. If there is an edge between node A and node B in π, it is one of the following categories:
   - $A \rightarrow B$
   - $A \circ\!\!\rightarrow B$
   - $A \circ\!\!-\!\!\circ B$
   - $A \leftrightarrow B$
3. There is at most one edge between any pair of vertices in π.
4. A and B are adjacent in π iff for every subset $\mathbf{Z}$ of $\mathbf{O} \setminus \{A, B\}$, G does not entail that A and B are independent conditioned on $\mathbf{Z} \cup \mathbf{S}$.
5. An edge between A and B in π is oriented as $A \rightarrow B$ only if A is an ancestor of B but not $\mathbf{S}$ in every DAG in **O-Equiv**(G).
6. An edge between A and B in π is oriented as $A* \rightarrow B$ only if B is not an ancestor of A or $\mathbf{S}$ in every DAG in **O-Equiv**(G).
7. $A * — * \underline{B} * — * C$ in π only if in every DAG in **O-Equiv**(G) either B is an ancestor of C, or A or $\mathbf{S}$. (Note that if the PAG does not contain $A * \rightarrow B \leftarrow * C$, then the underlining of B should be assumed to be present.)

   The reader is referred to (Spirtes et al., 1999) for further details and examples.

Faithful Data D over a set of observed variables $\mathbf{O}$ is said to be faithful to the generating causal structure G if the marginal and conditional probability distributions in D (according to some test "T") represent the dependencies and independencies in G. See also Section 2.4.

Definite noncollider (Spirtes et al., 2000, page 136) B is a definite noncollider on undirected path U iff *one* of the following conditions are met:

1. B is an endpoint of U.
2. There exist vertices A and C such that U contains one of the subpaths $A \leftarrow B* — *C$, $A* — *B \rightarrow C$, or $A* — *\underline{B}* — *C$, where $\underline{B}$ denotes that B is not a collider.

Possible-D-Sep(A, B, π) (Spirtes et al., 1999, page 227) We define **Possible-D-Sep**(A, B, π) where π represents a PAG as follows: If A and B are independent conditional on any subset of $\mathbf{O} \setminus \{A, B\} \cup \mathrm{S}$, then they are independent given some subset of **Possible-D-Sep**(A, B, π)

or **Possible-D-Sep**(B, A, π). A, B, and C for a *triangle* in a graph or a PAG iff A and B are adjacent, B and C are adjacent, and A and C are adjacent. V is in **Possible-D-Sep**(A, B) in π iff there is an undirected path U between A and B in π such that for every subpath $X* — *Y* — *Z$ of U either Y is a collider on the subpath, or X, Y, and Z form a triangle in π.

The following enumeration of the FCI algorithm is from (Spirtes et al., 2000, pages 144–145) (see also (Spirtes et al., 1999)).

3.1.2.4 FCI Pseudocode

A. Form the complete undirected graph Q on the set of variables $\mathbf{V}$.

B. $n = 0$.

repeat

repeat

select an ordered pair of variables X and Y that are adjacent in Q such that **Adjacencies**$(Q, X)\backslash\{Y\}$ has cardinality greater than or equal to n, and a subset $\mathbf{T}$ of **Adjacencies**$(Q, X)\backslash\{Y\}$ of cardinality n, and if X and Y are d-separated given $\mathbf{T}$ delete the edge between X and Y from Q, and record $\mathbf{T}$ in **Sepset**(X, Y) and **Sepset**(Y, X)

until all ordered variable pairs of adjacent variables X and Y such that **Adjacencies**$(Q, X)\backslash\{Y\}$ has cardinality greater than or equal to n and all subsets $\mathbf{T}$ of **Adjacencies**$(Q, X)\backslash\{Y\}$ of cardinality n have been tested for d-separation;
$n = n + 1$;

until for each ordered pair of adjacent vertices X, Y, **Adjacencies**$(Q, X)\backslash\{Y\}$ is of cardinality less than n.

C. Let F' be the undirected graph resulting from step B. Orient each edge as $\circ — \circ$. For each triple of vertices A, B, C such that the pair (A, B) and the pair (B, C) are each adjacent in F' but the pair (A, C) are not adjacent in F', orient $A* — *B* — *C$ as $A* \rightarrow B \leftarrow *C$ if and only if B is not in **Sepset**(A, C).

D. For each pair of variables A and B adjacent in F', if A and B are d-separated given any subset **T** of **Possible-D-SEP**$(A, B)\backslash\{A, B\}$ or any subset **T** of **Possible-D-SEP**$(B, A)\backslash\{A, B\}$ in F' remove the edge between A and B, and record **T** in **Sepset**(A, B) and **Sepset**(B, A).

E. The algorithm now proceeds to orient the edges by a complex set of rules. Some of these are similar to PC, but there are also some long-distance orientation rules (see (Spirtes et al., 2000, page 145) and (Spirtes et al., 1999) for the details).

Spirtes et al. state that the FCI algorithm outputs PAGs that are correct even when latent variables and selection bias may be present under assumptions of Markov, faithfulness, population inference and the assumption that there is a reliable statistical test for conditional dependence/independence (Spirtes et al., 1999).

The FCI algorithm can handle hidden variables and hence does not require the assumption of *causal sufficiency*. In the worst case FCI is exponential in the number of variables in the dataset. The number of conditional independence tests done by FCI grows exponentially as a function of the number of variables in the dataset. Apart from the increased running time of the algorithm, the higher order conditional independence tests are also unreliable because of sample size limits. Hence FCI is not suitable for datasets with large numbers of variables and high order dependencies (Spirtes, 2001).

Spirtes et al. have reported limited empirical results with the FCI algorithm cite[page 232–234]spirtes-meek-rich-99. Based on simulation data from the Alarm network (see Section 6.3.2) FCI output 62 ancestor relations with 100% accuracy and 1088 non-ancestor relations with an accuracy of 97%. A node A is an ancestor of node B iff there is a directed path from A to B.

3.1.2.5 Anytime FCI (AFCI) The *anytime* FCI algorithm (Spirtes, 2001) is an extension of FCI that takes an anytime approach to causal discovery to accommodate large datasets. This is accomplished by limiting the size of the conditioning set for performing the independence tests. At any time, this limit can be imposed on the outer loop of algorithm responsible for choosing this set and the algorithm is allowed to complete the subsequent

steps. Capping the size of the conditioning set makes the algorithm run faster. It also eliminates the higher order tests of independence that are likely to be unreliable. The "anytime" graph is correct with respect to a future output but could be less informative. Once the algorithm orients an edge it is never re-oriented in a future iteration. Proof of the anytime property of the algorithm is provided in (Spirtes, 2001).

3.1.3 GS Markov blanket algorithm

Researchers have proposed a constraint-based algorithm for induction of Bayesian networks by first identifying the *neighborhood* (Markov blanket) of each node (Margaritis & Thrun, 2000). The Grow-Shrink (GS) Markov blanket algorithm attempts to address the two main limitations of the PC and FCI algorithms—1. exponential time complexity and 2. higher order conditional independence tests (Margaritis & Thrun, 2000). However, it is still exponential in the size of the Markov blanket.

Test results on simulated data from artificial networks with a degree of five showed an edge error (failure of detection/wrong inclusion) of 20% and 30% error in specifying the directionality of edges correctly (Margaritis & Thrun, 2000). Since the Markov blanket algorithm does not make explicit claims about causal discovery, we will not further consider this algorithm.

Aliferis et al. have proposed HITON, an algorithm to determine the MB of an outcome variable (Aliferis et al., 2003). Tsamardinos et al. have described an algorithm called MMMB that discusses the possibility of interpreting the MB as direct causal relationships. However, the MMMB algorithm does not specifically distinguish between causes and effects of a node. The paper presents empirical results that compares well to the PC algorithm for small networks. The authors also show that the MMMB scales well for large networks with thousands of variables (Tsamardinos et al., 2003).

3.2 BAYESIAN NETWORK EQUIVALENCE

Before describing the score-based search methods, we introduce some definitions relevant to the search. We focus on the concept of equivalence relating to Bayesian networks. The following discussion of *equivalence* is based on (Heckerman, 1996; Heckerman et al., 1995).

3.2.1 Independence equivalence

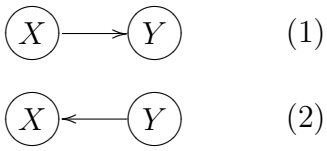


Figure 10: Two-variable independence-equivalent Bayesian networks

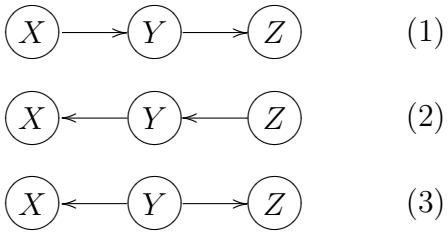


Figure 11: Three-variable independence-equivalent Bayesian networks

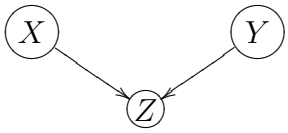


Figure 12: A "V" structure over variables X, Y, and Z.

Consider two Bayesian network structures B_{s1} and B_{s2}. B_{s1} and B_{s2} are said to be independence equivalent if they represent exactly the same conditional independence assertions

for $\mathbf{V}$, where $\mathbf{V}$ is the set of variables in each of B_{s1} and B_{s2} (Verma & Pearl, 1991). The two network structures in Figure 10 are independence equivalent. In particular they represent an assertion of no conditional independence. Likewise, all the three network structures in Figure 11 are also independence equivalent, asserting that X and Z are conditionally independent given Y.

Two network structures B_{s1} and B_{s2} are independence equivalent iff they satisfy the following conditions (Verma & Pearl, 1991):

1. B_{s1} and B_{s2} have the same set of vertices.
2. B_{s1} and B_{s2} have the same set of edges ignoring arc directions.
3. If there is a configuration such as $X \rightarrow Z \leftarrow Y$ and *no* arc between X and Y ("V" structure) in B_{s1}, the same pattern is present in B_{s2}.

A "V" structure over variables X, Y and Z is shown in Figure 12. There is a directed edge from X to Z and another directed edge from Y to Z. There is no edge between X and Y. A "V" structure contains a "collider" and the node Z is a collider in Figure 12.

3.2.2 Distribution equivalence

The concept of distribution equivalence is based on parameterization and is related to independence equivalence. Two network structures B_{s1} and B_{s2} over $\mathbf{V}$ are distribution equivalent with respect to a family of distributions $\mathcal{F}$ if they can represent exactly the same joint probability distributions for $\mathbf{V}$. This means that for every parameterization $\theta_{B_{s1}}$ of Bayesian network structure B_{s1}, there exists a parameterization $\theta_{B_{s2}}$ of structure B_{s2} such that $P(v|\theta_{B_{s1}}, B^h_{s1}) = P(v|\theta_{B_{s2}}, B^h_{s2})$, and vice versa. Here v is any variable such that $v \in \mathbf{V}$ and B^h_{s1} denotes the hypothesis that the joint probability distribution can be factored according to B_{s1}. There is an assumption of faithfulness here. Distribution equivalence implies independence equivalence but the converse is not necessarily true (Heckerman, 1996).

3.2.3 Hypothesis equivalence

Heckerman et al. (Heckerman et al., 1995) introduce the concept of *hypothesis equivalence* which states that if two networks B_{s1} and B_{s2} are distribution equivalent, then the hypothesis

associated with B_{s1} and B_{s2} are identical, i.e. $B^h_{s1} = B^h_{s2}$. If we consider the structures over variables X and Y given in Figure 10 and assume that $|X| = |Y| = 2$, it is easy to see that both the hypothesis $B^h_{X \to Y}$ and $B^h_{X \leftarrow Y}$ assert that the parameterization Θ_{XY} is unconstrained. Hence $B^h_{X \to Y} = B^h_{X \leftarrow Y}$. For a causal network, hypothesis equivalence does not necessarily imply causal hypothesis equivalence. Two network structures B_{s1} and B_{s2} propose the same causal hypothesis iff

1. B_{s1} and B_{s2} have the same set of vertices.
2. B_{s1} and B_{s2} have the same set of directed edges.
3. B_{s1} and B_{s2} are distribution equivalent.

For example, $B^h_{X \to Y}$ considered as a causal network implies that X causally influences Y, while $B^h_{X \leftarrow Y}$ is a different causal model in which Y causally influences X.

3.2.4 Likelihood equivalence

An assumption of *likelihood equivalence* is considered appropriate for learning causal networks in many domains. It implies that observational data cannot distinguish between distribution-equivalent network structures. Consider the two network structures B_{s1} $(X \to Y)$ and B_{s2} $(X \leftarrow Y)$. The lower case letters x and y refer to instantiations of the random variables X and Y respectively. The joint probability distribution of B_{s1} can be expressed as $P(x) \cdot P(y|x)$, and that of B_{s2} as $P(y).P(x|y)$. Based on the axioms of probability both these expressions can be written as $P(x, y)$. By the definition of conditional probability $P(x) \cdot P(y|x) = P(x, y) = P(y) \cdot P(x|y)$ and this shows that the parameters of B_{s1} can be inferred from that of B_{s2} and vice versa. Note that B_{s1} and B_{s2} have the same dependence/independence assertion i.e. $X \not\perp\!\!\!\perp Y$. We now look at B_{s1} $(X \to Y)$ and B_{s2} $(X \leftarrow Y)$ from a causal perspective. Assume that they have the same underlying joint probability distribution i.e. they are distribution equivalent. But they represent two distinct causal hypotheses:

1. X causes Y — (B_{s1})
2. Y causes X — (B_{s2})

Since observational data cannot distinguish causal hypothesis (1) from causal hypothesis (2), we would have to collect experimental data in such situations (Heckerman, 1995). But

observational data can be used to learn the set of directed edges that are common to *all* the members (structures) that are distribution equivalent or in other words the directed edges of the essential graph that represents the joint probability distribution over the observed variables.

The parameter independence assumption states that the parameters associated with each variable in a BN structure are independent (global parameter independence). Likewise, the parameters associated with each instantiated configuration (state) of the parents of a variable are also independent (local parameter independence). See (Heckerman, 1995) for more details.

The multinomial assumption states that each instance in a dataset D is drawn independently from the parameter Θ of the population. For the Bayesian network represented in Figure 13 with three variables X, Y, and Z, this parameter can be represented as θ_{xyz}. Note that for multinomial models (models that assume a multinomial distribution) that also assume parameter independence such as the one shown in Figure 13, distribution equivalence $\Leftrightarrow$ independence equivalence i.e. one implies the other (Heckerman, 1995).

Heckerman et al. (Heckerman et al., 1995) define likelihood equivalence as follows: Given two network structures B_{s1} and B_{s2} such that $P(B_{s1}^h|\mathcal{K}) > 0$ and $P(B_{s2}^h|\mathcal{K}) > 0$, if B_{s1} and B_{s2} are distribution-equivalent, then $P(\Theta_{\mathbf{V}}|B_{s1}^h, \mathcal{K}) = P(\Theta_{\mathbf{V}}|B_{s2}^h, \mathcal{K})$ where $\mathcal{K}$ represents background knowledge, $\mathbf{V}$ is the set of variables in B, $\Theta_{\mathbf{V}}$ is the joint distribution over the variables in $\mathbf{V}$.

3.3 SCORE BASED LEARNING ALGORITHMS

While constraint-based methods make qualitative or categorical dependence/independence statements, score-based methods make probabilistic inferences about marginal and conditional independencies in a domain based on data and prior knowledge (Heckerman et al., 1999).

Most of the score-based methods described in the literature adopt a global search strategy

in which the goal is to learn a global Bayesian network model of the data. The K2 algorithm of Cooper and Herskovits uses a greedy search strategy to identify the parents of a node and scores the resulting DAGs using the K2 scoring measure. Since the K2 score does not necessarily assign the same score to DAGs that are likelihood equivalent, Heckerman et al. introduced the BDe metric that assigns the same score to members of likelihood equivalent classes of Bayesian networks. See (Cooper & Herskovits, 1992; Heckerman et al., 1995) for more details. More recently researchers have extended the line of research of learning causal Bayesian networks from larger datasets (Friedman et al., 1999; Friedman & Koller, 2000; Friedman et al., 2000; Pe'er et al., 2001). Using a Bayesian network framework Pe'er et al. present a method to learn subnetworks of gene interactions using perturbed gene expression data of S. cerevisiae (Pe'er et al., 2001). Their method initially uses global search to identify highly probable features (for example, edges) and then builds subnetworks using local search. A key way in which the method of Pe'er et al. differs from BLCD in the requirement of perturbed (experimental) data for assigning a causal semantics to the discovered subnetworks.

In their study with data obtained from single cells perturbed with molecular interventions, Sachs et al. learned causal protein-signaling networks using a global Bayesian network algorithm. Seventeen directed arcs were predicted by their model of which fifteen had been already reported in literature. The authors were able to confirm the remaining two using experimental methods. When they used only observational data, their algorithm output only ten undirected arcs of which eight were expected based on domain knowledge and two were unexplained. The method also missed eleven arcs that were expected based on prior knowledge (Sachs et al., 2005). The method of Sachs et al. is not suitable for inferring causal relationships from observational data alone due to the following limitations of their approach:

1. They were inducing Bayesian networks in general without explicitly modeling causal relationships.
2. They were not looking for Y structures which are needed for discovering causal structures from observational data using a Bayesian network framework unless additional assumptions (for example, causal sufficiency or prior knowledge) are made.

In contrast BLCD and its variants are designed specifically to discover causal relationships from observational data.

In his review Nir Friedman proposes probabilistic graphical modeling as a sound paradigm for inferring biological pathways from high throughput data such as gene expression datasets (Friedman, 2004). Though the review cites different studies that use the graphical modeling framework for inferring gene regulatory networks, the review does not provide an explicit causal semantics for probabilistic modeling using causal Bayesian networks, which is needed for providing a causal interpretation to the inferred cellular networks.

We do not discuss these algorithms in further details since they do not claim to discover *causal* Bayesian networks from observational data.

An interesting algorithm for learning Bayesian networks using a global score-based approach has been developed by Moore and Wong. The algorithm introduces a new search operator called optimal reinsertion (OR). On each step a node is labeled as the target node. All incoming and outgoing arcs of this target node are removed and the node is reinserted with the "optimal" combination of incoming and outgoing arcs. The process is repeated with all nodes taking turns as the target node multiple times until no step changes the Bayesian network structure (Moore & Wong, 2003). Even though this is not a causal discovery algorithm, we use this algorithm for comparison purposes after suitable modification to output causal relationships.

Before we move on to a discussion of local search-based methods, we discuss the challenges of hidden variables and also the role played by instrumental variables in causal discovery.

3.3.1 Hidden variables

Here we introduce some of the possible approaches for handling these unobserved variables. The following summary is based on (Heckerman et al., 1999). A variable is hidden when its values are absent in the dataset for all instances. The exact computation of the parameters of a model M in the presence of hidden variables is intractable and typically one of the approximation methods enumerated below could be employed:

1. Monte-Carlo methods such as Gibbs sampling

2. Gaussian approximation

3. Maximum a posteriori (MAP) and maximum likelihood approximations using the EM algorithm

The interested reader is referred to (Heckerman et al., 1999; Geman & Geman, 1984; Dempster et al., 1977) for details.

Tian and Pearl describe methods for validating causal models in the presence of hidden variables. They show that non-independence constraints, also referred to as functional constraints could be used to distinguish models that belong to the same independence equivalence class (Tian & Pearl, 2002).

3.3.2 Instrumental variables

We now touch upon the role of instrumental variables (IV) in causal discovery. Instrumental variables are exogenous variables i.e. variables external to a system that is being studied. For example, to ascertain the causal influence of smoking on lung cancer from observational data, a variable such as tobacco advertisement or cigarette price can be leveraged to play the role of an IV variable (Bowden & Turkington, 1984; Pearl, 1994).

Based on inequality constraints induced by observed variables when the independence relationships of two models $\mathcal{M}_1$ and $\mathcal{M}_2$ are the same, Pearl has proposed methods to test if a particular variable can play the role of an instrument (Pearl, 1995). (Pearl, 1994) discusses a role for mediating instrumental variables in addition to the use of traditional IV for understanding causality. A mediating IV is internal or endogenous to the system and the causal influence of a variable X on variable Y is mediated through such a variable Z.

Let us now consider another situation where a randomized controlled trial was done to ascertain the value of a particular treatment A versus treatment B (placebo). When subject compliance is imperfect the effect of treatment cannot be quantified correctly even when sample size is large. Chickering and Pearl propose a system that combines Bayesian learning and Gibbs sampling to derive the posterior distribution of the treatment effect in the population (Chickering & Pearl, 1996).

3.4 LOCAL SEARCH BASED METHODS

Local search methods do not strive to build a full CBN. LCD (Cooper, 1997) and LCD variants (Mani & Cooper, 2001) are constraint-based algorithms that employ local search. They take as input a dataset and available prior knowledge and then output purported causes of the form variable X causally influences variable Y. These algorithms have a time complexity that is $O(mn^2)$ in the worst case where m is the number of records in the database and n is the number of variables in the database. Another algorithm that is related to LCD is the Silverstein algorithm. The LCD algorithm is described in Section 3.4.1, the Silverstein algorithm in Section 3.4.2 and the LCD variants in Section 4.1.1 and Section 4.1.2.

3.4.1 LCD

In this section, we describe the LCD algorithm (Cooper, 1997) on which several variant algorithms are based. LCD assumes the following:

Assumption 1: The causal Markov condition.
Assumption 2: The causal faithfulness condition.

Before introducing the LCD algorithm in detail, we discuss two additional assumptions that LCD makes apart from the causal Markov and causal faithfulness assumptions described in Section 2.4. In causal discovery, we do not know the probabilistic relationships among variables precisely, because we only have a finite amount of data. Thus, we introduce the following assumption:

Assumption 3: The statistical testing assumption. A statistical test performed to determine independence (or alternatively dependence) given a finite dataset will be correct relative to independence (dependence) in the joint probability distribution that is defined by the causal process under study.

That is, we assume our statistical test gives valid independence and dependence results among the variables being measured (and about which we wish to discover causal relationships). In general, the greater the number of records in a dataset, the more likely it is that the statistical testing assumption will hold. But even at very large sample sizes, spurious correlations may be seen, as the results of statistical tests may be sensitive to factors such as

violations of distributional assumptions, and measurement errors etc., which erode the validity of statistical tests (Glymour et al., 1999, page 328). The reader is referred to (Glymour & Cooper, 1999, chapters 8–11) for a discussion of this and other related issues.

In addition, LCD makes the following assumption:

Assumption 4: Given measured variables W, X, and Y, if X causes Y, and X and Y are not confounded, then one of the causal networks in Figure 14 must hold.

Assumption 4 means that W is not causally influenced by X or by Y. As we discuss in later sections, in our experiments we chose W so that this assumption is tenable. We also sometimes refer to this fourth assumption as the "W" variable assumption. W plays a role similar to an instrumental variable (see Section 3.3.2).

Before introducing the LCD algorithm in more detail, we define some key terms. Let $\text{Independent}_T(A,\ B)$ denote that A and B are independent according to test T applied to our dataset. Let $\text{Independent}_T(A,\ B \text{ given } C)$ denote that A and B are independent given C, according to T. Finally, let $\text{Dependent}_T(A,\ B)$ denote that A and B are dependent according to T[4]. These independence and dependence tests are labeled as given below for easy reference.

Test_1. $\text{Dependent}_T(W,\ X)$
Test_2. $\text{Dependent}_T(X,\ Y)$
Test_3. $\text{Dependent}_T(W,\ Y)$
Test_4. $\text{Independent}_T(W,\ Y \text{ given } X)$

If all four of these tests are satisfied then LCD outputs that X causally influences Y. The first network in Figure 14 violates Test_1, and thus, LCD is unable to detect that X causally influences Y in such situations. Under Assumptions 1 through 3, the other three networks in Figure 14 satisfy Test_1 through Test_4. In (Cooper, 1997), it is shown that if X and Y are confounded, then one or more of the four tests will be violated. As an example, Figure 15 shows an important case in which X and Y are confounded by a hidden variable H. For this causal network, it follows from Assumption 2 that W and Y will be dependent given X, and thus, Test_4 will fail.

[4]Although the three tests in this paragraph should technically be distinguished from each other by using separate labels, such as *T1*, *T2*, and *T3*, for simplicity of notation we use a single label T.

To summarize, under Assumptions 1 through 4, when X causally influences Y and these two variables are unconfounded, the four tests hold (unless W and X are independent). Conversely, when X and Y are confounded (or when W and X are independent), one or more of the four tests will fail. From these propositions, we can conclude that if the four tests hold, then one of the three causal networks (2,3, or 4) in Figure 14 must hold, and thus, we can determine that X causally influences Y and the two variables are unconfounded. Since LCD outputs pairwise causal relationships, using a triplet WXY and evaluating it based on the outcome of Test_1 – Test_4, it is a local anytime algorithm.

3.4.2 Silverstein algorithm

Silverstein and others have recently described a constraint-based algorithm (Silverstein et al., 2000) that is related to LCD. They discuss their algorithm in the context of market basket analysis and empirically demonstrate the applicability to large datasets including textual data. Their algorithm is based on the following two rules:

CCC causality rule If there are three variables W, X and Y such that the pairs WX, XY and WY are dependent, and W and Y become independent given X, then one of the causal models shown in Figure 16 hold. This is valid if there are no additional variables (measured or unmeasured) that are causing two or more of W, X, Y. If an additional assumption (apart from the Markov and faithfulness assumptions) is made that W has no causes among the measured variables, then the above conditions imply the model (1) in Figure 16 will hold even in the presence of confounding variables.

CCU causality rule Assume that the following relationships hold for the three variables W, X and Y—the pairs (W, X) and (X, Y) are dependent, and (W, Y) is independent. When conditioned on X, the (W, Y) pair become dependent. Then in the absence of confounding variables, it is possible to conclude that W and Y cause X.

Though LCD uses the CCC causality rule, it makes the following modified assumption. Instead of assuming that W has no causes, it just assumes that it is not causally influenced by either X or Y. Hence LCD concludes that X causes Y but it makes no conclusion that W causes X.

The CCU rule is problematic. Even if we make the assumption that there are no unmeasured variables, using this rule we cannot conclude that both W and Y cause X if there are other measured variables that are common causes of (W and X) or (W and Y). On the other hand, even allowing for confounders (hidden and measured), we can conclude that X *does not* cause W or Y. Figure 17 shows a confounder for the pair (W, X) and the pair (X, Y). In other words we can make an acausal discovery but not a causal one using the CCU rule.

3.4.3 Instrumental variable (IV) algorithm

The IV algorithm (Spirtes & Cooper, 1999) is a particular instrumental variable algorithm (Bowden & Turkington, 1984; Pearl, 1995) that is similar to the LCD algorithm. It takes as input prior knowledge in the form of instrumental (exogenous) variables, a dataset D and outputs pairwise causal relationships of the form variable A causes variable B.

The IV algorithm evaluates triplets of variables E, A, B in which E is an instrument and A and B are other measured variables. The selection of the E, A, B triplet might be based on a time ordering $E < A < B$, where $X < Y$ means that X precedes Y in time. The IV algorithm tests for statistical dependence between pairs E, A, A, B, and E, B. If all the pairwise dependence tests are positive based on a user-defined threshold, a conditional independence (CI) test (Independent$(E, B|A)$) is performed. If the CI test is also positive, the posterior probability of the model $E \rightarrow A \rightarrow B$ is computed using the BDe scoring measure over all DAG models in which E is exogenous. If no DAG has a higher posterior probability than $E \rightarrow A \rightarrow B$, then the IV algorithm outputs that A causes B. See (Spirtes & Cooper, 1999) for more details. Using a database of pneumonia patients, the authors compared the output of the IV algorithm with the evaluation of physicians about the "causality" of the relationships. The study failed to show a good correspondence between the output of IV and the assessment of the physicians in terms of causality. The paper suggests modifications to the algorithm to improve performance.

To the best of our knowledge there are no other Bayesian local causal discovery algorithms described in the literature.

3.5 MDL METHODS

In this section we introduce the concept of Occam's razor, the MDL principle, and then describe methods of learning Bayesian networks from data using this principle.

The concept of Occam's razor is that less complex models should be preferred to more complex models, all else being equal. Consider the following two causal models described over three observed variables X, Y and Z.

Given a joint probability distribution of the three variables based on observational data, assume that both of these models (Model 1 and Model 2, figure 18) are equally likely. As they encode the same probabilistic independence relationships assuming the Markov and faithfulness conditions ($\text{Dep}(X, Z)$, $\text{Dep}(Y, Z)$, $(X \perp\!\!\!\perp Y)$ and $\text{Dep}(X, Y|Z)$) they are likelihood equivalent. However, using the concept of Occam's razor model (1) would be preferred as it is simpler and can be expressed more compactly.

The minimum message length (MML) or the minimum description length (MDL) principle (as it is more popularly known) has been developed from fundamental principles of information coding theory (Rissanen, 1978; Mitchell, 1997; Wallace & Korb, 1999; Lam & Bacchus, 1994). We now describe the connection between Bayes rule and the MDL principle. The following discussion is based on (Mitchell, 1997). Suppose we have some evidence (data) for a hypothesis $h \in \mathcal{H}$ and we also have a prior probability of h. Using Bayes rule (theorem) we can compute the posterior probability of h as follows:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{3.1}$$

If we have a set of hypotheses $\mathcal{H}$, and we want to find the hypothesis h that is most probable (the maximum a posteriori or MAP hypothesis), we could take the following approach.

$$\begin{aligned} h_{MAP} &\equiv argmax_{h \in \mathcal{H}}\, P(h|D) \\ &= argmax_{h \in \mathcal{H}}\, \frac{P(D|h)P(h)}{P(D)} \\ &= argmax_{h \in \mathcal{H}}\, P(D|h)P(h) \end{aligned} \tag{3.2}$$

The final step leaves out the term $P(D)$ because it is a constant relative to different hypotheses h.

The MAP of equation 3.2 can be expressed in log terms as follows.

$$h_{MAP} = argmax_{h \in \mathcal{H}} \ (log_2 P(D|h) + log_2 P(h)) \tag{3.3}$$

Taking the negative log and using *argmin* we obtain

$$h_{MAP} = argmin_{h \in \mathcal{H}} \ (-log_2 P(D|h) - log_2 P(h)) \tag{3.4}$$

From coding theory we know the following principle. Optimal coding that assigns the minimum number of bits to a message can be realized by using shorter codes for messages with higher probability. Shannon showed that such an optimal coding scheme uses $-log_2 P_i$ bits for message i that has probability P_i of occurring. We can interpret equation 3.4 based on the above result from coding theory as follows:

1. $-log_2 P(D|h)$ is the description length of the training data D given hypothesis h under its optimal encoding. If specified in terms of number of bits, $L_{bits}(D|h)$ denotes the number of bits required in an optimal coding scheme for data D, assuming that both the sender and receiver know the hypothesis h.
2. $-log_2 P(h)$ is the description length of h under the optimal encoding for the hypothesis space $\mathcal{H}$. If specified in terms of number of bits, $L_{bits}(h)$ denotes the number of bits required for h assuming an optimal coding scheme for the hypothesis space $\mathcal{H}$.

   Equation 3.4 can now be written in terms of length specified as number of bits.

$$h_{MAP} = argmin_{h \in \mathcal{H}} \ (L_{bits}(D|h) + L_{bits}(h)) \tag{3.5}$$

The MDL principle selects the hypothesis h that minimizes the number of bits required to represent h and $(D|h)$. Note though that the obtained minimization is an upper bound assuming optimal representation for concepts h and $(D|h)$.

Using the MDL principle a test instance can be classified probabilistically by combining the predictions of all the hypotheses multiplied by their posterior probabilities. The probability of class $c_j \in C$ given the test instance can be computed as follows:

$$P(c_j|D) = \sum_{h \in \mathcal{H}} P(c_j|h)P(h|D) \tag{3.6}$$

The new instance is assigned the class c_j that maximizes $P(c_j|D)$.

The first application of the MDL principle for Bayesian network learning was by Lam and Bacchus (Lam & Bacchus, 1994). Since identifying the network model minimizing the representation of $h + (D|h)$ is not feasible by searching all possible networks, they provide a heuristic search algorithm to reduce the computational complexity. The MDL approach will favor networks with lesser number of parents for each node as representation complexity (length) increases with greater size of the parent set (for example, the conditional probability tables will be larger). For representing $(D|h)$, they use the Kullback-Leibler cross entropy measure—the cross entropy between the distribution defined by the model and the true distribution. They approximate the true distribution by decomposing the joint probability distribution specified by a Bayesian network into a product of lower-order marginals. The choice of the specific lower-order marginals will determine the accuracy of the estimate.

Even though in the general case of Bayesian networks this cross entropy estimation is computationally intractable, the paper discusses an extension of the approach taken by Chow and Liu for learning decision trees using local computation (Chow & Liu, 1968) to reduce the computational complexity. By limiting the number of parents a node can have to k, such that $k \ll n$, where n is the total number of variables, the complexity of their Bayesian network learning algorithm is $O(mn^4)$ where m is the total number of cases in the dataset (Lam & Bacchus, 1994). The authors evaluated their algorithm based on simulated data from artificial networks that they created as well as known synthetic networks such as the Alarm network. They report that the algorithm gives a trade-off between accuracy and complexity of the learned structure—a simpler network was always recovered when the algorithm failed to recover the original network.

Wallace and Korb also describe a framework for learning linear causal models by applying the MML principle (Wallace & Korb, 1999). They state that real-world data is usually characterized by the following properties that can be used for causal discovery:

1. A temporal order for the variables that determines the direction of causal arcs.
2. Linear causal influences among variables.
3. Endogenous variance of each variable. This refers to unexplained differences or variation seen in a variable.

However, if prior knowledge in the form of a temporal order for the variables is not available, they make the assumption that all temporal orders are equally likely *a priori*. They introduce the concept of a *totally ordered model* or TOM. The idea of TOM is based on MML equivalence sets—sets of models with similar regression parameters. Given a fixed ordering of the variables and a set of direct causal links, the task then is to identify the set containing the maximum likelihood regression parameter estimates for the given data. This set is called a TOM. The task of causal discovery can be stated as the goal of identifying the best total ordering of the variables plus the best set of causal relationships.

Their MML algorithm uses a Monte carlo sampling technique incorporating a version of the Metropolis algorithm (Chib & Greenberg, 1995). They derive the joint probability of TOM and data, then make use of Fisher information to identify the TOM of interest and output the set of causal arcs and their parameters. The authors report that the performance of their MML algorithm is as good or better than Tetrad II particularly when no prior knowledge was made available to the programs.

Note that the MDL/MML principle as an approximation of the MAP hypothesis is dependent on the optimal representation (encoding) for hypothesis h and $(D|h)$. We have to show that the size of the encoded hypothesis h is $-log_2 P(h)$ and that we can represent the encoding of D given h with $-log_2 P(D|h)$. Such a representation necessitates knowledge of all prior probabilities $P(h)$ and $P(D|h)$ for $h \in \mathcal{H}$. Moreover, in certain situations a human expert might be able to specify a compact representation using relative probabilities of the hypothesis rather than the full specification of the probability of each hypothesis. Hence the application of the MDL to arbitrary concept codings may not result in identification of good models (Mitchell, 1997).

3.6 LEARNING AND REPRESENTING LOCAL PARAMETER STRUCTURE USING OTHER FORMALISMS

This section introduces local parameter structure representation. Note that this is different from the local causal discovery framework that we introduced where the goal is to learn

causal models over a subset of variables.

Friedman and Goldszmidt (Friedman & Goldszmidt, 1999) discuss methods to represent and learn *local parameter structure*[5] in the context of learning Bayesian networks from data. They focus on learning and representing conditional probability distributions (CPDs) compactly using *default tables* and probability trees. Default tables explicitly represent only the conditional probabilities that are different, the other rows (entries) get collapsed to one single *default* row. The authors argue that learning these reductionist local parameter structures results in two improvements to the induced models: (1) the parameters are more reliable because of better estimates and (2) the global search procedure for a better scoring network is also benefited by local parameter reduction.

Chickering et al. (Chickering et al., 1997) describe a method of representing CPDs as decision graphs that are more general than decision trees. The paper also derives a Bayesian score for structures containing these graphs. Empirical results show that greedy search over structures with decision graphs yield higher scoring structures when compared to decision trees and complete conditional probability tables.

3.7 HYBRID METHODS OF LEARNING NETWORKS

We now describe three algorithms that combine two different methods: (1) information theoretic plus constraint based (Cheng et al., 1997), and (2) Bayesian scoring plus constraint-based (Singh & Valtorta, 1995; Spirtes & Meek, 1995; Dash & Druzdzel, 1999). Even though they are BN learning algorithms per se and not causal discovery algorithms, we summarize them to show that the different approaches that we introduced earlier could be combined. The first two hybrid algorithms (Cheng et al., 1997; Singh & Valtorta, 1995) do not assume causal sufficiency whereas (Spirtes & Meek, 1995; Dash & Druzdzel, 1999) make explicit assumptions of causal sufficiency.

Cheng et al. (Cheng et al., 1997) describe an algorithm for learning Bayesian networks from data based on ascertaining dependence/independence among variables using mutual

[5]We use local parameter structure to distinguish it from the "local" search concept in the LCD framework.

information and conditional independence (CI) tests. They assume that an ordering of the variables is possible and also that the data will facilitate reliable CI tests. Using pairwise mutual information, a draft DAG is created in the first step and then refined by addition/deletion of edges based on CI tests. The use of pairwise mutual information is the information theoretic component of their method. A key feature of the algorithm is identification of minimal conditioning sets (d-separation sets) for the CI tests. This refers to the minimal set of nodes $\mathbf{Z}$ required to d-separate a pair of nodes X and Y. Even though their procedure used a greedy heuristic that is not guaranteed to find minimal conditioning sets, the authors state that for the Alarm network it actually found more than 200 conditioning sets—all of them minimal. The algorithm requires CI tests of $O(n^2)$ time complexity where n is the total number of measured variables being modeled. A variant of the algorithm that does not require a node ordering is less efficient with a time complexity of $O(n^4)$ in terms of CI tests. In the worst case, when all the CI tests require conditioning on all the other nodes, the time complexity is $O(n^2 r^n)$ where r is the number of levels (states) of a variable and n, the number of variables. Thus both these algorithms have worst-case search time complexities that are exponential in n.

Singh and Valtorta (Singh & Valtorta, 1995) describe a hybrid Bayesian network learning algorithm that combines constraint-based and Bayesian scoring methods of learning from data. This algorithm, named the CB algorithm uses CI tests in the first phase to order the variables and then uses the K2 algorithm (Cooper & Herskovits, 1992) in the second phase to learn the network. The two phases are executed in an iterative fashion starting with 0th order CI tests, then 1st order and on to higher order tests until the stopping criteria for the algorithm is met. One stopping criterion is non-improvement of network score compared to the previous iteration. They also discuss simulated annealing techniques to surmount local minima yielding better scoring networks in a subsequent iteration. For sparse graphs, lower order CI tests are sufficient and hence the CB has polynomial time complexity in such cases. The worst case complexity is exponential since high order CI tests will be required to handle dense networks.

Spirtes and Meek (Spirtes & Meek, 1995) discuss an algorithm that is a combination of the constraint-based PC algorithm (Spirtes et al., 1993) and a Bayesian search algorithm that

does not require a node ordering. They call this algorithm PC + GBPS (since it performs a greedy pattern search). Given as input an empty pattern or a pattern from the output of a PC algorithm run, GBPS adds edges to the pattern in a greedy fashion till no further addition increases the score. It then starts removing edges until no further edge deletion improves the score. The PC algorithm can be used efficiently to provide an initial pattern as input to GBPS as it can get a close-to-correct pattern in a relatively short period of time. The initial pattern generation is mostly based on lower order conditional independence tests performed by the PC algorithm that are computationally efficient. This step considerably shortens the search time of GBPS that is used to refine the initial pattern.

Dash and Druzdzel (Dash & Druzdzel, 1999) describe a hybrid *anytime* algorithm that first uses constraint-based methods to narrow the search for Bayesian network structures and then applies Bayesian scoring to select a model. They discuss EGS (denotes essential graph search) and its variant EGS/GS that are extensions to the PC algorithm (Spirtes et al., 1993).

Since EGS and EGS/GS are extensions of the PC, they still have worst time complexities that are exponential in the largest indegree in the output graph. For example, if the largest in-degree is k and the number of variables is n, the time complexity is $O(n^k)$.

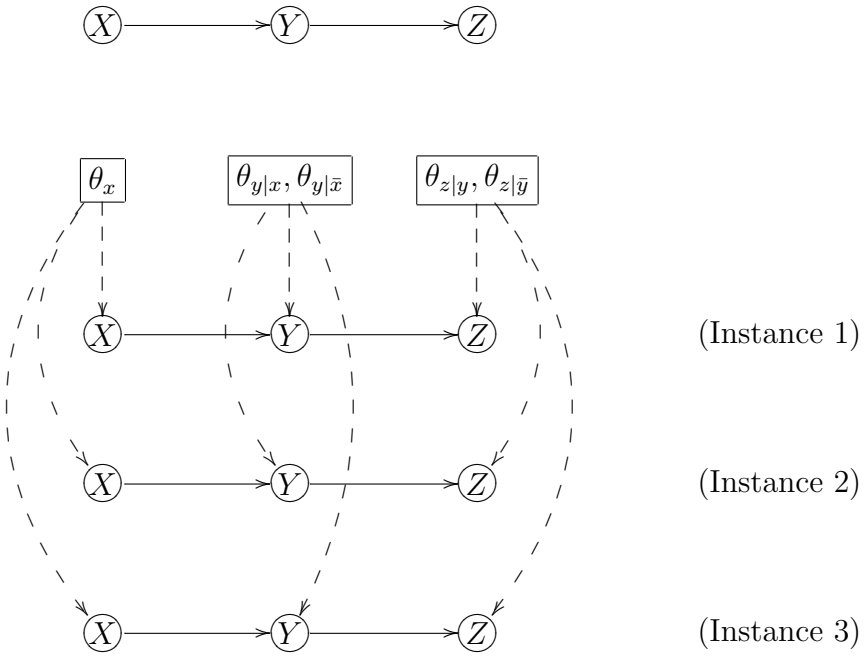


Figure 13: Bayesian network structure $X \rightarrow Y \rightarrow Z$ annotated with the conditional independence relationships resulting from the multinomial sample and parameter independence assumption [Modified from (Heckerman et al., 1995)]

.

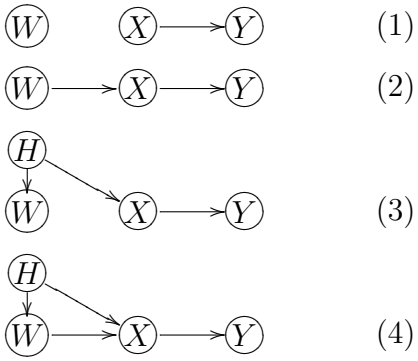


Figure 14: Causal models in which X causes Y; H denotes a hidden variable(s).

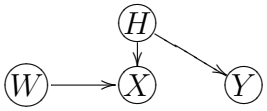


Figure 15: Causal model in which W causes X, and X and Y are dependent due to confounding by a hidden variable(s) represented by H.

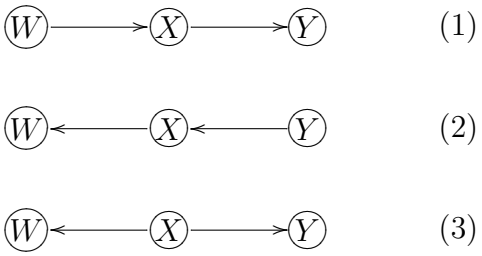


Figure 16: Three causal models for variables W, X and Y

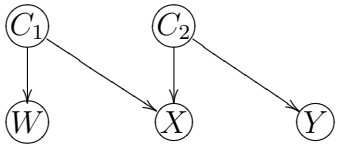


Figure 17: A model that satisfies the CCU rule and is confounded

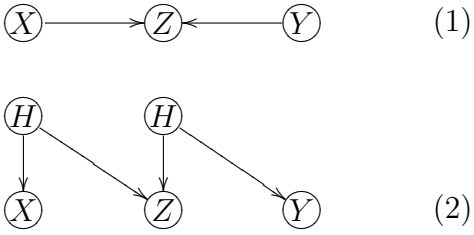


Figure 18: Two causal models with equivalent independence relationships

4.0 PRIOR WORK

In this section we discuss preliminary studies that are part of this dissertation research and focus on constraint-based local causal discovery (LCD) algorithm variants. To recapitulate, in the local causal discovery framework, the goal is to discover submodels (for example, pairwise causal relationships) of the full causal network using a local search methodology that considers only a small subset of observed variables (for example, triplets). A variable not known to be causally influenced by any of the other observed variables in the database is considered to be an *instrumental variable* for causal discovery using LCD. The LCD algorithm takes as input a dataset D over a set of observed random variables, available prior knowledge in the form of *instrumental variables* ("W" variable), and then outputs purported causes of the form "variable X causally influences variable Y". LCD is described in Section 3.4.1. The goal of this prior work described in this chapter has been to apply LCD to real-world and simulated datasets in order to critically evaluate and subsequently extend the LCD framework. A description of the LCD variant algorithms is provided in Section 4.1 and summaries of our experimental results in different domains are given in Section 4.2.

4.1 LCD VARIANTS—ALGORITHM DESCRIPTION

In this section, we describe the LCD variants that are based on the LCD algorithm described in Section 3.4.1. These algorithms also make the following four assumptions:

Assumption 1: The causal Markov condition
Assumption 2: The causal faithfulness condition
Assumption 3: The statistical testing assumption
Assumption 4: The "W" variable assumption

4.1.1 LCDa, LCDb and LCDc

The motivation for considering variants of LCD came from our observation that in all the false positive causal output from LCD based on the Alarm network (Beinlich et al., 1990) data, the independence test (Test_4)[1] was returned as positive when it should have failed (Mani & Cooper, 2001). The dependence tests (Test_1 through Test_3) did not fail. This led us to explore more stringent tests of independence. For example, performing an increased number of independence tests for the same XY pair using different W nodes might improve independence testing, resulting in a more accurate assessment of the causal influence of X on Y. This was our working hypothesis in the design of LCD variants LCDa, LCDb, and LCDc.

We now describe these variants in greater detail. The LCDa, LCDb and LCDc algorithms apply Test_1 through Test_4 in exploring a database for possible causal relationships. These variants make use of more than one W variable to perform additional independence tests when feasible. If we have a $\mathbf{W}$ set consisting of two or more W variables, the WXY triplets are generated as follows: Let $\mathbf{V}$ be the set of all variables and $\mathbf{Z} = \mathbf{V} \setminus \mathbf{W}$. Create WXY triplets satisfying the following constraints.

1. $W \in \mathbf{W}$.
2. $X \in \mathbf{Z}$ and $Y \in \mathbf{Z}$ and $X \neq Y$.

The algorithms LCDa, LCDb and LCDc perform Test_1 through Test_3 (see Section 3.4.1) for all such triplets WXY in the database (see Section 3.4.1 for a description of the tests). Test_1 through Test_3 output many triplets W_iXY such that for the same pair XY there often is more than one W_i. In such situations Test_4 could be taken as positive if it is satisfied for any *one* such triplet (LCDa), satisfied by any *two* such triplets (LCDb) or satisfied by *all* such triplets (LCDc). Note that if there is only one triplet for a pair XY, LCDa, LCDb and LCDc perform Test_4 on just that one triplet WXY. The Independence and Dependence tests described in (Cooper, 1997) can be used. Both tests have $O(m)$ time complexity, where m is the number of records (cases) in the database. If all four tests are passed, LCDa, LCDb and LCDc output that X causally influences Y and the two variables are unconfounded

[1] These statistical tests are described in Section 3.4.1.

(under Assumptions 1–4), and the probability distribution of Y given X is displayed.

4.1.2 Contextual causal influences—LCDm

For each causal relationship of the form $X \rightarrow Y$ we identify a set of higher order (multivariate) or contextual influences acting on the effect node (Y) and incorporate them in the model as described below. We use "higher order", "multivariate", "covariate" and "contextual" interchangeably to refer to the set of variables which modify the causal influence of a variable X on variable Y. Node B is said to be a *covariate* of node A if A and B are probabilistically dependent. Currently these contextual influences are considered one at a time and denoted by the letter M.

Figure 19 shows graphically the influence of a variable M on the effect node Y. The first four situations can be identified by the statistical tests presented in Table 1, as explained below.

A higher order influence M will satisfy Test_5 and either Test_{6a} or Test_{6b} if the generating causal structure can be represented by models (1), (2), (3) or (4) in Figure 19. Note that model (5), model (6) and model (7) will satisfy Test_5 but not Test_{6a} or Test_{6b} and hence such models will be rejected by LCDm. The tests Test_{6a} and Test_{6b} are important in identifying models such as (7), where conditioning on M induces a dependency between X and Y. In model (6), the M variable acts as a mediating variable for the causal influence of X on Y and hence conditioning on M does not induce a dependency between X and Y. However, since model (6) fails both Test_{6a} and Test_{6b}, it will be excluded.

Procedure LCDm
$\mathbf{V}$ Set of random observed variables
$W, X, Y \in \mathbf{V}$
$\mathbf{V}_1 = \{\mathbf{V} \setminus W\}$;

Foreach $X \in \mathbf{V}_1$
DO
    $\mathbf{V}_2 = \{\mathbf{V}_1 \setminus X\}$;
    Foreach $Y \in \mathbf{V}_2$
    DO
        Test_1. $W \not\perp\!\!\!\perp X$; IF False break;
        Test_2. $X \not\perp\!\!\!\perp Y$; IF False break;
        Test_3. $W \not\perp\!\!\!\perp Y$; IF False break;

```
        Test_4. W ⫫ Y | X; IF False break;
        Procedure MV(V, W, X, Y);
    OD
OD

Procedure MV (Input V, W, X, Y)
DO
    V' = V \ {W, X, Y};
    Foreach M ∈ V'
    DO
        Test_5. Y ⫫̸ M; IF False break;
        Test_6a. X ⫫ M | Y;
        Test_6b. X ⫫ M;
        If Test_6a or Test_6b is True
        DO
            Output X causally influences Y in context M;
            Output P(Y | X, M);
        OD
    OD
OD
```

We refer to each M variable as m_i. The modified LCD (LCDm) initially checks for the causal influence of X on Y using the four tests (Test_1 through Test_4 described in Section 3.4.1). If all four tests are satisfied, then it performs Test_5, Test_{6a}, and Test_{6b} to identify the M variables. LCDm outputs the m_i variables for each causal influence of the form X causes Y. The distribution of Y given X and m_i is also estimated from the dataset and output (Mani, 2000).

The LCD variants (LCDa, LCDb, LCDc, and LCDm) like LCD evaluate pairwise causal relationships and output them. These variant algorithms also fit well into the *anytime* framework.

4.2 LCD VARIANTS—RESULTS AND DISCUSSION

We first describe our prior work using simulated data (Mani & Cooper, 2001). Evaluation of learned causal output is difficult, due to lack of a gold standard in real-world domains.

Therefore, we used simulated data from a known causal network in a medical domain—the Alarm network (see Section 6.3.2). For causal discovery we used LCDa, LCDb and LCDc. Using the simulated Alarm dataset as input, LCDa had a false positive rate of 0.09, LCDb 0.08, and LCDc 0.04. All the algorithms had a true positive rate of about 0.27. Most of the false positives occurred when a causal relationship was confounded. Among causally confounded pairs[2], LCDc output as causal only those pairs that had very weak confounding. The strength of confounding can be assessed by error measurements proposed to predict the distribution of Y given that X is observed and also to predict the distribution of Y given that X is manipulated (Cooper & Yoo, 1999). These measurements were adapted in our study of LCDa, LCDb, and LCDc using the Alarm network (Mani & Cooper, 2001).

LCDb and LCDc make use of more independence tests when there is more than one X variable for a pair Y, Z. This results in elimination of pairs with relatively higher confounding. Qualitatively we identified the causally-confounded patterns which were output by LCDa but not by LCDb and LCDc. Figure 20 shows a representative example.

In this example LCDa output Y causally influences Z, while LCDb and LCDc did not. The independence test—(IND $(x1, Z|Y)$) was positive while (IND $(x2, Z|Y)$) was negative. Since LCDa requires only one positive independence test, it output $Y \rightarrow Z$. This can be explained by the fact that with $x2$ confounding is more direct and local. The $x1$ confounding path $(x1 \rightarrow Y \leftarrow x2 \rightarrow Z)$ is longer than the $x2$ confounding path $(x2 \rightarrow Z)$.

Since we are interested in discovering unconfounded causal relationships while keeping false positives to a minimum, the property of outputting only pairs that had very weak confounding along with causal pairs makes LCDc an attractive algorithm for causal discovery. The interested reader is referred to (Mani & Cooper, 2001) for more details.

We also applied the LCD algorithm to investigate factors causally influencing infant mortality in the United States using the US Linked Birth/Infant Death dataset for the year 1991 that had more than four million records and about two hundred variables for each record. Our study sample consisted of 41,000 records that we randomly selected from the whole dataset. LCD output nine "potential" causal relationships. Eight out of the nine

[2]A pair XY is causally confounded if there is a directed path connecting X and Y, plus a node Z that has a directed path to X and a different directed path to Y

relationships appear to be plausibly causal (Mani & Cooper, 1999).

With a view to test the applicability of the local causal discovery techniques to textual data, we performed a preliminary study using intensive care unit (ICU) discharge summaries of about 1600 patients (Mani & Cooper, 2000). Medical records usually incorporate investigative reports, historical notes, patient encounters or discharge summaries as textual data. Identification of the causal factors of clinical conditions and outcomes making use of this textual information (data) might be helpful in formulating better management, prevention and control strategies for the improvement of health care. Using the words that occur in the discharge summaries as attributes for input, LCD output three purported causal relationships when a threshold level of 0.9 was used for ascertaining dependence and independence.

One relationship discovered was that *alcoholic* causally influences *cirrhosis*. A closely related one is that *alcohol* causally influences *cirrhosis*. These relationships are well-known in the medical literature. Table 2 gives the probability distribution of *cirrhosis* given *alcoholic*. The third relationship that *portal* causally influences *cirrhosis* seems to be plausible only in the reverse direction, assuming *portal* denotes *portal hypertension*. It is possible that this false positive output was due to subtle confounding, which could be eliminated if the confounders (e.g., alcoholic) are considered. Using a threshold of 0.8 and lower gave five relationships that appear unreliable including for example that "ascitis causes cirrhosis". Hence, I did not analyze those results further. I did not extend this line of research from textual data, as the focus of my dissertation research was generation of causal models from coded data.

As described in Section 4.1.2, LCDm first uses the framework of LCD to discover pairwise causal influences and then identifies a set of variables that play a modifying role. For example, we could ask questions such as—"how is the causal influence of *infant birth weight* on *infant mortality* modified by *delivery conductor* (MD, nurse, midwife)?". Note that LCDm basically depends on LCD for first identifying the causal influence of X on Y. What LCDm does is subsequently determine how this causal influence is modified by the covariates of Y. We have developed an algorithm that incorporates the covariates of Y in the causal discovery process in an interesting way (see Section 5.1).

Table 3 summarizes the causal output at the dependence and independence threshold

level of 0.9. The first seven entries seem plausible. We focus on one (Birth-weight) to illustrate the clinical utility of looking for multivariate influences. Figure 21 and Table 4 give the probability distribution of *Infant Mortality* given *Birth-weight*. Figure 22 and Table 5 show the influence of the variable *Hemoglobinopathy in Mother* on the causal influence of *Birth-weight* on *Infant Mortality*.

Figure 22 demonstrates the multivariate effect of birth weight and maternal disease (hemoglobinopathy). But the deleterious effect of the presence of both hemoglobinopathy and low birth weight has to be interpreted with caution because of the small sample sizes. Further study using larger samples is required to confirm this effect.

We have extended this prior work in the following general direction. Keeping the framework of local causal discovery (i.e., using local search), we have developed new algorithms that we hypothesize will improve causal discovery performance, by retaining existing algorithmic efficiency while increasing the true positive rate and decreasing the false positive rate of purported non-confounded causal relationships that are output. Specifically we extend the LCD framework in the following ways:

1. Instead of a constraint-based approach, a Bayesian methodology is employed for causal discovery.
2. The methods are designed to discover observed direct causes. For example, if the causal effect of X_1 on Y is mediated through X_2, we will be able to report that X_2 causally influences Y.
3. We relax the requirement of inputting W variables.
4. The new framework has the flexibility to incorporate various types of prior knowledge.

These extensions form the core of my dissertation research to develop a Bayesian local causal discovery (BLCD) framework which is described in Chapter 5.

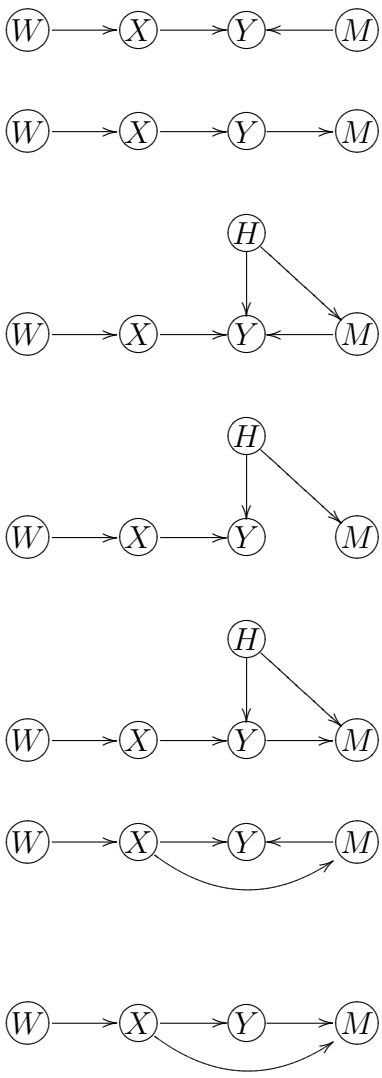


Figure 19: Selected causal models in which W causes X, and X causes Y; M acts as a covariate of Y. H denotes a hidden variable(s).

Table 1: LCDm statistical tests for the models shown in Figure 19

Test Name	Test Description	$\mathcal{M}$ (1)	$\mathcal{M}$ (2)	$\mathcal{M}$ (3)	$\mathcal{M}$ (4)	$\mathcal{M}$ (5)	$\mathcal{M}$ (6)	$\mathcal{M}$ (7)
Test_5	$\text{Dependent}_T(Y, M)$	+	+	+	+	+	+	+
Test_{6a}	$\text{Independent}_T(X, M \mid Y)$	−	+	−	−	−	−	−
Test_{6b}	$\text{Independent}_T(X, M)$	+	−	+	+	−	−	−

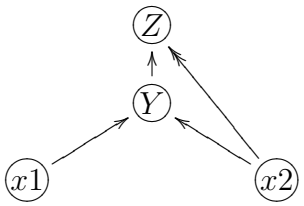


Figure 20: A causally-confounded pattern output by LCDa, but not by LCDb or LCDc. A double arrow denotes a path length greater than one.

Table 2: Conditional probability table of cirrhosis given alcoholic

	Alcoholic	
Cirrhosis	Absent	Present
Absent	0.96*	0.61
Present	0.04	0.39

*The probability that cirrhosis is absent *given* that alcoholic is absent.

Table 3: LCDm output—X causally influencing Y, and the number of multivariate influences for $X \rightarrow Y$

Cause (X)	Effect (Y)	# of M nodes
Maternal weight gain	Infant Mortality	6
Adequacy of prenatal care	Infant Mortality	1
Gestational duration	Infant Mortality	9
Birth-weight	Infant Mortality	2
Multiple Pregnancy	Infant Mortality	11
One-minute Apgar Score	Infant Mortality	7
Five-minute Apgar Score	Infant Mortality	6
Assisted ventilation	Infant Mortality	11

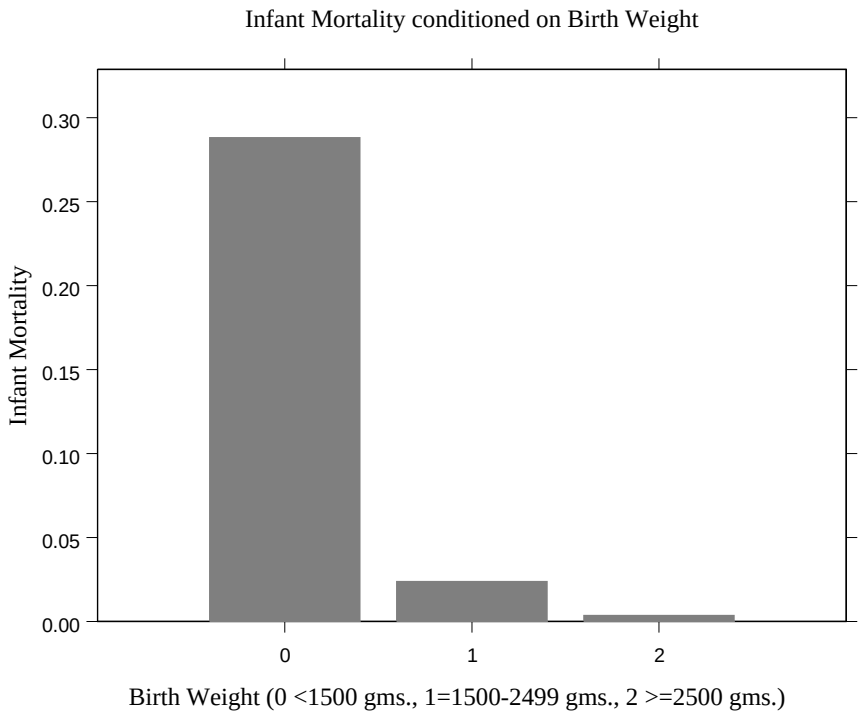


Figure 21: Influence of Birth Weight on Infant Mortality

Table 4: Infant outcome given infant birth weight

Birth Weight	Infant outcome at one year		
	Died	Survived	MR
<1500 gms.	141	350	0.288*
1500–2499 gms.	57	2394	0.024
$\geq$ 2500 gms.	128	38036	0.003

MR—Mortality Rate

*The probability that Infant outcome at one year equals Died *given* that Infant Birth Weight is <1500 grams.

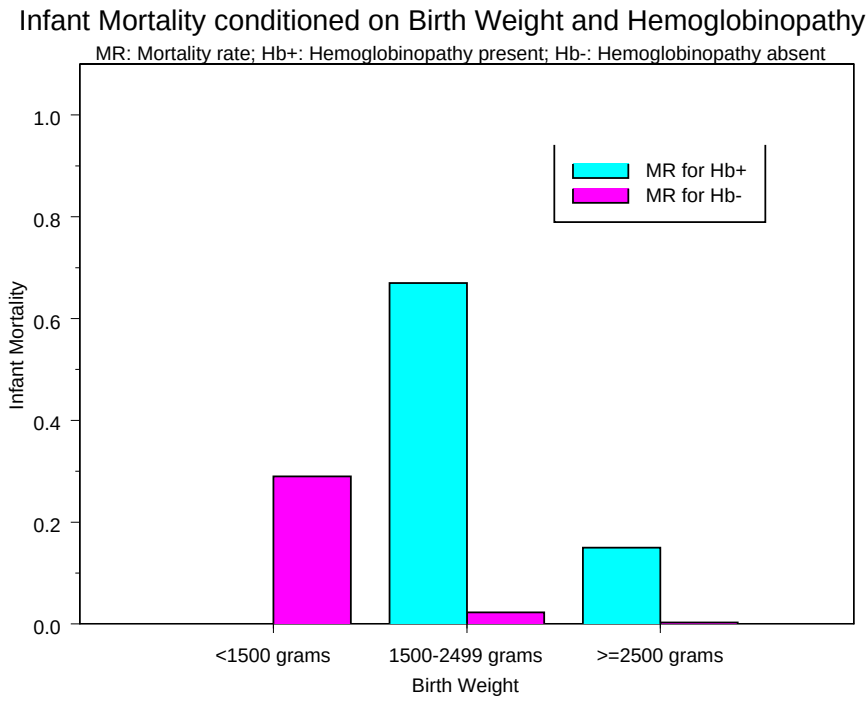


Figure 22: Multivariate Influence on Infant Mortality

Table 5: Infant outcome given infant birth weight and maternal disease

Birth Weight	Hb	Infant outcome at one year		
		Died	Survived	MR
<1500 gms.	Yes	0	0	-
<1500 gms.	No	127	303	0.29*
1500–2499 gms.	Yes	1	0	0.67
1500–2499 gms.	No	51	2178	0.023
$\geq$ 2500 gms.	Yes	1	10	0.15
$\geq$ 2500 gms.	No	120	35145	0.003

Hb—Hemoglobinopathy in mother, MR—Mortality Rate

*The probability that Infant outcome at one year equals Died *given* that Infant Birth Weight is <1500 grams and Hemoglobinopathy is absent.

5.0 ALGORITHMIC METHODS

In this chapter we introduce our framework for discovering causal relationships efficiently from observational data. The chapter is organized as follows. Section 5.1 presents the Bayesian local causal discovery (BLCD) framework for causal discovery and the BLCD algorithm based on that framework. In Section 5.2, we describe extensions to the basic BLCD algorithm.

5.1 THE BAYESIAN LOCAL CAUSAL DISCOVERY (BLCD) FRAMEWORK AND BLCD ALGORITHM

In the BLCD framework, we use causal Bayesian networks to represent causal relationships among model variables. To recapitulate, a causal Bayesian network (CBN) is a Bayesian network in which the directed arcs represent causal influences between the nodes. See Section 2.2 for details.

The following notational convention will be used for the description of the BLCD algorithms. Sets of variables are represented in bold and upper case, random variables by upper case letters italicized and lower case letters will be used to represent the value of a variable or sets of variables. When we say $\mathbf{X} = \mathbf{x}$, we mean an instantiation of all the variables in $\mathbf{X}$, while $X = x$ denotes that the variable X is assigned the value x. Graphs are denoted by calligraphic letter, such as $\mathcal{G}$ or upper case letters such as G or M.

We introduce a Bayesian local causal discovery algorithm (BLCD) that conjectures causal relationships between pairs of variables that have no common causes (confounders). Instead of using constraint-based independence and dependence tests, we score the models by a

Bayesian method. This confers the following advantages:

1. Allows informative causal priors to be incorporated.
2. Provides a quantitative posterior assessment of causality, based on prior belief and data.
3. Does not require a special instrumental variable.

BLCD assumes the following:

Assumption 1: The causal Markov condition
Assumption 2: The causal faithfulness condition

We now present the definitions of "V" structure and "collider" here for ease of reference. A "V" structure over variables W_1, W_2 and X is shown in Figure 23. There is a directed edge from W_1 to X and another directed edge from W_2 to X. There is no edge between W_1 and W_2. A "V" structure contains a "collider" and the node X is a collider in Figure 23. Since there is no arc between W_1 and W_2, X is also termed as an *unshielded* collider. Figure 24 shows a model in which there is an arc between W_1 and W_2, and thus X is a *shielded* collider. BLCD requires that a node X be an unshielded collider in order to discover the causal effects of X.

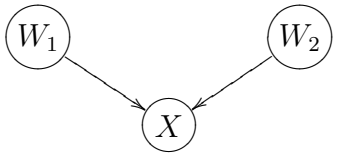


Figure 23: A "V" structure—X is a collider in this Figure

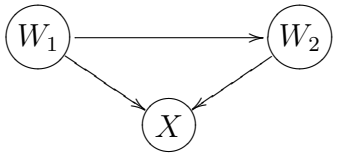


Figure 24: A "shielded" collider—X is a shielded collider in this example

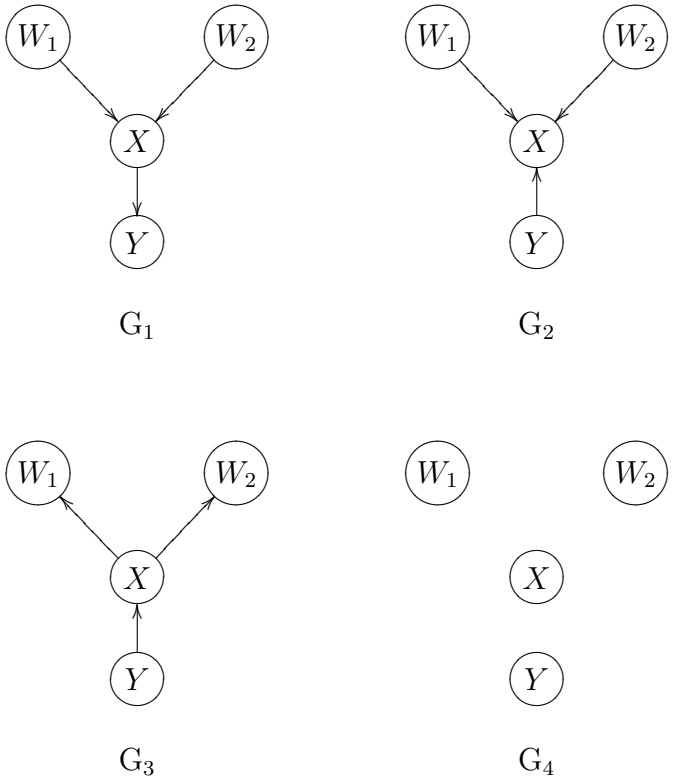


Figure 25: Several causal models that contain four nodes out of the possible 543 models.

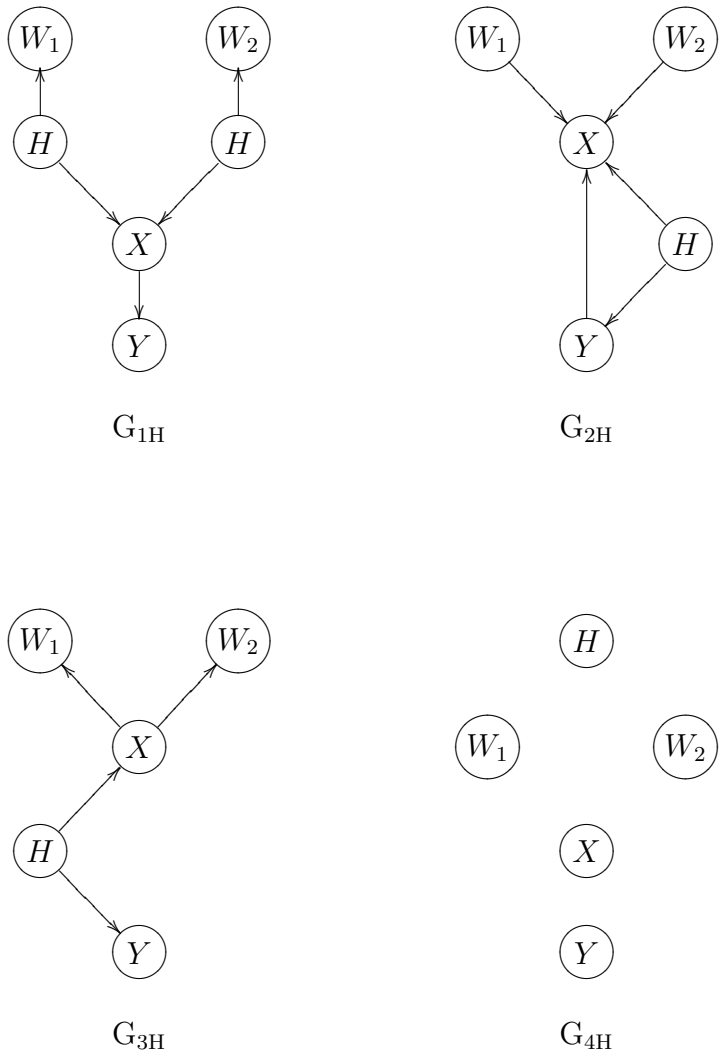


Figure 26: Four causal models containing one or more hidden variables that represent the same independence relationships of the corresponding models shown in Figure 25. A hidden variable is represented with the letter H.

5.1.1 Y structure

In this section we introduce the concept of a Y structure. Let $W_1 \longrightarrow X \longleftarrow W_2$ be a V structure. Note that X is an unshielded collider in this V structure since there is no arc between W_1 and W_2. If there is a node Y such that there is an arc from X to Y, then the nodes W_1, W_2, X and Y form a Y structure. A Y structure has interesting dependence and independence properties.

If W_1, W_2, X, Y form a Y structure over a set of four variables $\mathbf{V}$ and the Y structure is represented by G_1 (see Figure 25), then there is no other structure $G_{i(i\neq 1)}$ over $\mathbf{V}$ that is in the same independence equivalence class as G_1. In other words, if a Y structure is learned from data, the arc from X to Y represents an unconfounded causal relationship. Since G_1 also has the same I-map as G_{1H} (see Figure 26), the arcs $W_1 \longrightarrow X$ and $W_2 \longrightarrow X$ cannot be interpreted as causal relationships.

We also refer to the Y structure represented by G_1 in Figure 25 as a Y skeleton. We now introduce the concept of *Mconnected* which is needed for understanding the various types of Y structures. Two variables X and Y are said to be Mconnected iff they satisfy the following conditions:

1. X and Y are d-connected by one or more measured variables.
2. There is no arc between X and Y.

The different types of Y structures encountered in a BN are defined below: Note that the notation $A \,\square\, B$ means that there is no arc between A and B.

Y skeleton: W_1, W_2, X, Y form a Y skeleton iff

1. W_1 and W_2 are parents of X
2. X is a parent of Y
3. $W_1 \,\square\, W_2$
4. $W_1 \,\square\, Y$
5. $W_2 \,\square\, Y$

Global Y: W_1, W_2, X, Y form a Y skeleton, there may or may not be measured confounders for X and Y, and W_1 and W_2 may or may not be Mconnected (d-connected through a measured variable).

Mshielded Y: W_1, W_2, X, Y form a Y skeleton, and W_1 and W_2 are Mconnected.

Unshielded Y: W_1, W_2, X, Y form a Y skeleton, and W_1 and W_2 are not Mconnected.

Unconfounded Y: W_1, W_2, X, Y form a Y skeleton, X and Y are unconfounded.

Using the BLCD search strategy, for each pair of nodes X and Y where X is a collider, the probability of $X \rightarrow Y$ will be derived under assumptions. We illustrate this first using a hypothetical domain with four discrete random observed variables—W_1, W_2, X, and Y (See Figure 25). The model G_1 has the "Y" structure format. A "Y" structure is required to infer pairwise causal relationships in an unconfounded way from observational data making just the two basic assumptions (causal Markov and causal faithfulness) for causal discovery.

Figure 25 also shows models G_2, G_3, and G_4 from our four variable domain. The four models shown are a few examples of the 543 potential CBN models for a four variable domain (Cooper, 1999, page 43). When BLCD scores a model it also implicitly scores models with hidden variables that map to it. Figure 26 shows some hidden variable models that map to the corresponding models without hidden variables. For example, when the model G_1 is scored, the model G_{1H} that maps to it is also scored. Thus, the score for a CBN in BLCD is a score for one measured-only model (a model without hidden variables) and many hidden variable models (models containing four observed and one or more hidden variables).

The following equation provides a lower bound on the probability of an unconfounded causal relationship between X and Y:

$$P(X \rightarrow Y|D) \geq \frac{\text{Score}(G_1|D)}{\sum_{i=1}^{543} \text{Score}(G_i|D)} \tag{5.1}$$

where D is the dataset.

Note also that the 543 CBNs can be partitioned into equivalence classes containing one or more CBNs. The 543 CBNs can be grouped into 185 equivalence classes (Gillispie & Perlman, 2002). Figure 27 shows four CBNs belonging to the same equivalence class. Making use of this equivalence property it is possible to compute the sum score of all the 543 models (the denominator in the right hand side of Equation 5.1) by scoring one representative model

each from the equivalence class and multiplying by the corresponding number of models. The sum score can be computed using the following equation:

$$\sum_{i=1}^{543} \text{Score}(\text{G}_i|\text{D}) \quad = \quad \sum_{j=1}^{185} \text{Score}(\text{E}_j|\text{D}) * |(\text{E}_\text{j})| \tag{5.2}$$

where D is the dataset, 185 is the total number of equivalence classes, E_j is a representative member of the jth equivalence class and $|(\text{E}_j)|$ gives the number of CBNs in the jth equivalence class.

However, the current implementation of BLCD simply uses Equation 5.1 to compute the sum score.

We now describe the matrix method that BLCD uses to generate the 543 models. The matrix method is based on the following combinatoric formalism. Let **F** represent the set of four nodes. For each node $A \in \mathbf{F}$ we define a potential parent set **S** such that $\mathbf{S} = \mathbf{F} \setminus A$. Let **Q** represent the power set of **S**. Each member of **Q** is a potential parent set for a node A (see Table 6).

Using the representation in Table 6 we assign numbers $0, 1, 2, \ldots, 7$ for the members of **Q**. We represent all possible *unconstrained directed graphs* by four octal digits and they are numbered $0000, 0001, \ldots, 7776, 7777$. Note that in these unconstrained directed graphs if A is a parent of B we allow B to be a parent of A. In other words for each node $A \in \mathbf{F}$ we assign the parent set $\text{Pa}(A) \in \mathbf{Q}$ without any restrictions. From a set **F** with elements W_1, W_2, X, Y we thus create 4^8 unconstrained directed graphs. Three representative graphs are shown in Figure 28. Eliminating graphs with cycles we obtain 543 DAGs.

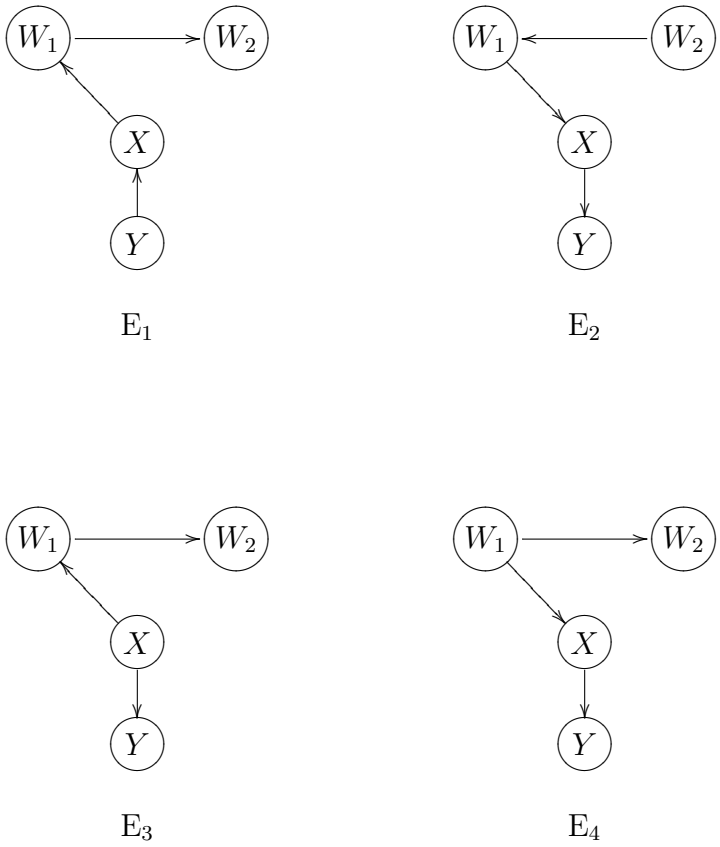


Figure 27: Four causal models out of the 543 that belong to the same equivalence class.

Table 6: DAG generation from a four node set **F**

Notation	Representation
F	$\{W_1, W_2, X, Y\}$
A	X
S for X	$\{W_1, W_2, Y\}$
Q	$\{\ \{\}, \{W_1\}, \{W_2\}, \{Y\}, \{W_1, W_2\}, \{W_1, Y\}, \{W_2, Y\}, \{W_1, W_2, Y\}\ \}$

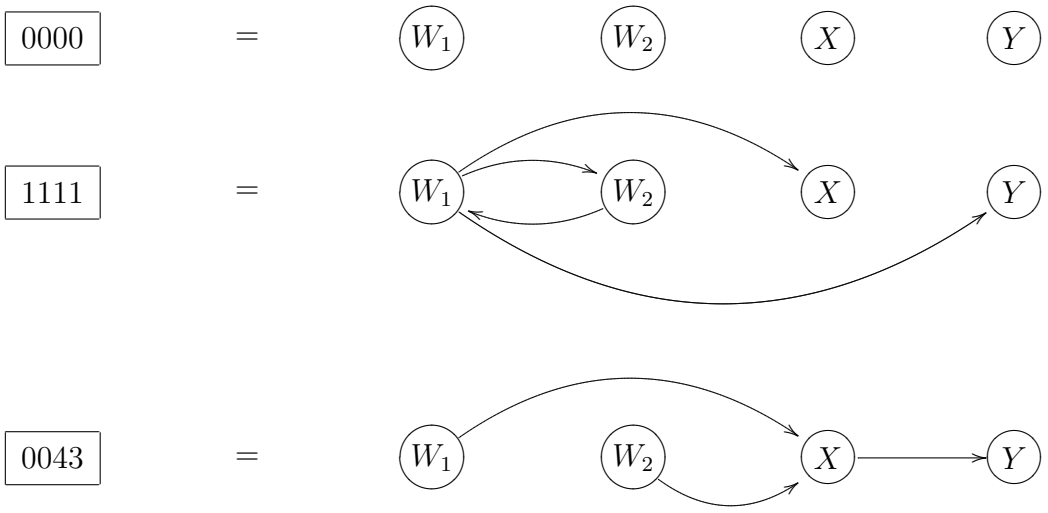


Figure 28: Three unconstrained directed graphs and their codes. The code 0043 represents the Y structure.

5.1.2 Scoring the DAGs

We score each node in **F** with a member from **Q** creating a 2D array of 32 entries (4 columns $\times$ 8 rows) which I will call SCORETABLE. Using the index i $(0 \ldots 3)$ for columns and the index j $(0 \ldots 7)$ for rows we can access the 32 scores and compute the scores for the 543 DAGs. We provide the pseudocode below:

SCOREHASH: hashtable;
DAGS: array[0 . . 542] of string; Each string is made up of 4 octal digits.
STRING: array[0 . . 3] of octal digit;
SCORETABLE: array[0 . . 7, 0 . . 3] of real;
i, j, digit: integer;
dagscore: real;
foreach STRING in DAGS
DO
    dagscore = 1.0;
    $i = 1$;
    foreach digit in STRING
    DO
        j = digit;
        dagscore = dagscore $*$ SCORETABLE$[i, j]$;
        $i++$;
    OD
    SCOREHASH{STRING} = dagscore;
OD

Note that it is possible to mark the DAGs that are representative of an equivalence class, determine the number of members of each such equivalence class and speed up the computation of the sum of the scores of all the 543 DAGs in Equation 5.1.

5.1.3 Scoring Measure

The Score function assigns a score to a model that represents the probability of the model given data and prior knowledge. For scoring the DAGs we used the Bayesian likelihood equivalent (BDe) metric (Heckerman et al., 1995) that is given below:

$$P(\text{S,D}) \quad = \quad P(\text{S}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{5.3}$$

where:

- $P(\text{S})$ is the prior probability of the CBN S.
- n is the number of nodes (variables) in the CBN.
- Γ is the gamma function.
- q_i is the number of unique instantiations of the parents of node i that is realized in database D. If node i has no parents, then $q_i = 1$.
- r_i denotes the number of discrete values of node i.
- N_{ijk} is the number of instances in D that node i has value k and the parents of i have the instantiation denoted by j.
- N_{ij} is the number of instances in D that the parents of node i have the instantiation denoted by j.
- α_{ijk} can be interpreted as the prior number of samples for which node i has value k and the parents of i have the instantiation denoted by j.
- α_{ij} can be interpreted as the prior number of samples for which the parents of node i have the instantiation denoted by j.

  Note that by definition:

$$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \text{ and } N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{5.4}$$

We used non-informative structural and parametric priors for scoring the CBNs. To be more specific, we assumed that all the 543 CBNs are equally likely and used 1/543 as the structural prior for each CBN ($P(\text{S}) = 1/543$). For the parametric priors we used an equivalent sample size of 1. Thus α_{ij} was assigned $1/q_i$ and α_{ijk} was assigned $1/r_i q_i$.

The 543 structures cover the space of CBN models exhaustively and no other structure has the same dependence/independence properties as G_1, i.e., there is no other CBN in the four node domain in the independence equivalent class of G_1. In the large sample limit the posterior probability of G_1 will be greater than the other 542 models if indeed (1) X causally influences Y in an unconfounded manner, (2) X is an unshielded collider of W_1 and W_2 in the distribution of the causal process generating the data, and (3) the Markov and Faithfulness conditions hold.

The heuristic methodology to select the tetrasets consisting of W_1, W_2, X and Y is given below. The method involves identification of the Markov Blanket of each random variable. As pointed out in Section 3.1.3 Aliferis et al. have proposed HITON, an algorithm to determine the MB of an outcome variable (Aliferis et al., 2003). It would be interesting to use HITON in step 1 of BLCD. We derive the MB of a node (designated as **B**) by a greedy heuristic search and refer to it as the *Procedure MB*. The set **B'** $\subseteq \mathbf{V} \setminus X$ that maximizes the score for the structure **B'** $\rightarrow X$ based on a greedy forward search as described in (Cooper & Herskovits, 1992) is initially identified. This is followed by a one step backward greedy search that prunes **B'** to yield set **B** $\subseteq$ **B'** that maximizes the score for the structure **B** $\rightarrow X$.

This set is updated using the following rule: If X is in the MB of Y, but Y is not in the MB of X, add Y to the MB of X. We refer to this rule as the MB *update* rule.

5.1.4 BLCD steps

For each node $X \in \mathbf{X}$ (**X** denotes the set of all random observed variables in the dataset) **DO**

1. **Derive the Markov Blanket.** Derive the Markov Blanket of X using the *Procedure MB*. Let **B** denote the MB of X. An user defined upper limit $\mathcal{L}$ is enforced for the cardinality of **B** s.t. $|\mathbf{B}| \leq \mathcal{L}$.
2. **Update B.** Apply the MB *update* rule. The update is performed such that $|\mathbf{B}| \leq \mathcal{L}$.
3. **Pick W_1, W_2, and Y.** Obtain all possible distinct triplets (sets of three nodes) from **B**. Add X to each triplet to get sets of four variables. We refer to each set of four variables

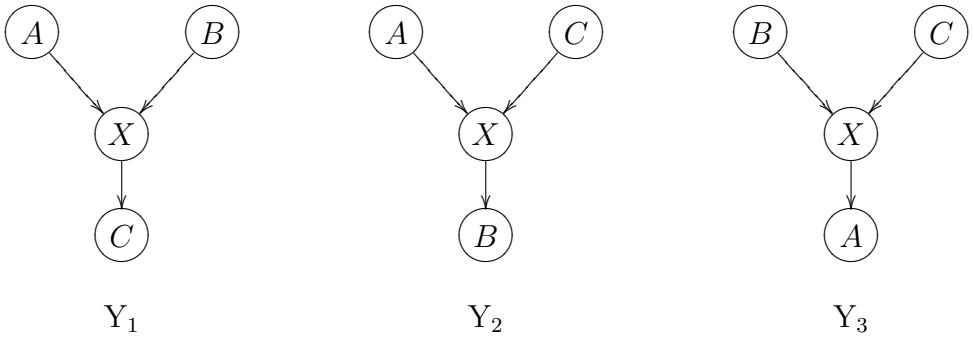


Figure 29: Node X and three nodes from the Markov Blanket of X give rise to three "Y" patterns—Y_1, Y_2, and Y_3.

as a tetraset $\mathbf{T}$. Since we are focusing on the MB of X, X is an essential element of $\mathbf{T}$. Note that each tetraset can give rise to 3 "Y" patterns where the X variable is a cause and each of the other three variables are potential effects (see Figure 29).

4. **Derive P**$(X \rightarrow Y|D)$: For each of the 3 "Y" patterns, the probability of $X \rightarrow Y$ is derived using Equation 5.1.

5. **Generate output**: If $P(X \rightarrow Y \mid D) > t$, where t is a user-set threshold, then output $X \rightarrow Y$ as a purported, unconfounded causal relationship.

OD

By setting t close to 1.0, we avoid false positives, which is an important goal. In causal datamining we do not want to overwhelm the user with many false positives. We would like to trade off recall (number of true causal relationships output over the total number of true causal relationships) for precision or positive predictive value (number of true causal relationships over the total number of relationships output). In other words we want to improve the signal to noise ratio in the output and the goal is to pick just the signals (true causal relationships).

5.1.5 Time complexity of BLCD

We assume here that the number of levels (states) of any of the variables in $\mathbf{V}$ is bounded by some constant. We also limit the number of elements in the MB of a variable X. This is done in Steps 1 and 2 of the BLCD algorithm. During the forward greedy search of step 1, a limit on the number of parents that can be added to a node X is set. If the limit is reached in Step 1, Step 2 is skipped for the node X. The *update* in Step 2 is done respecting the limit. Let this MB cardinality limit be r. This naturally limits the number of unique trisets (set of three variables representing $W1$, $W2$, and Y) that can be chosen from r. Let this number be s. Note that $s = O(\frac{1}{2}\binom{r}{3})$, where the $\frac{1}{2}$ is there because we do not distinguish between $W_1 = A$ and $W_2 = B$ versus $W_1 = B$ and $W_2 = A$, $\forall A, B \in$ MB. Each triset with the inclusion of X will result in a tetraset. We score the CBN models using a Bayesian scoring metric (BDe metric) that has a time complexity of $O(m)$ where m is the size of the dataset (number of instances). Each tetraset has 543 CBNs and hence the time complexity for scoring each tetraset is $O(m)$. The time complexity for scoring the tetrasets derived from the MB of one variable is thus $O(ms)$, and over all the variables it is $O(msn)$, where n the total number of variables in $\mathbf{V}$.

Now let us consider the time complexity of deriving the MB. Since we limit the number of elements in the MB of a node to r, the time complexity of deriving the MB of a node is $O(mnr)$, and over all the nodes it is $O(mn^2r)$. The time complexity of BLCD is thus $O(mn^2r) + O(msn) = O(mn(nr + s))$.

BLCD can be implemented as an *anytime* algorithm that outputs purported causes as they are found, if we skip Step 2 of the BLCD algorithm that updates the MB of a node. In particular, to output the purported effects of a node X, requires only the MB of X, data on both X and the variables in the MB of X.

5.1.6 Proof of correctness of BLCD

The two key steps of the BLCD algorithm are the following:

1. Identification of the MB of a node X.
2. Estimating the probability of the Y structures in the MB.

The reader is referred to Appendix D for theorems and proofs related to MB, and to Appendix C for Y structure related theorems and proofs.

5.1.7 Incorporating prior knowledge in BLCD

In this section we discuss how prior knowledge of a domain could be incorporated in the BLCD framework. A variety of techniques have been described in the literature for incorporation of prior knowledge for Bayesian networks (Heckerman, 1996; Buntine, 1991). The priors could be specified for whole CBNs, or perhaps more feasibly, the prior of each arc could be specified by an expert and an independence assumption made. One modular approach for specifying structure priors is to assign a probability to each of the three possible relationships between each pair of variables A and B:

1. $P(A \longrightarrow B)$
2. $P(A \longleftarrow B)$
3. $P(A \; . \; . \; B)$ (No edge between A and B)

As these three relationships are mutually exclusive and exhaustive the structure prior of any CBN can be derived by assuming independence of the pairwise priors and thereby taking the product over all the pairwise priors:

$$P(\mathrm{B}_S) \;\; = \;\; \prod_{A,B \in \mathbf{V}} P(\mathrm{R}(A,B)) \tag{5.5}$$

where B_S is a BN structure and $P(\mathrm{R}(A,B))$ denotes relationship between nodes A and B in B_S. This approach is a modification of the method provided in (Buntine, 1991).

However, the current version of BLCD has implemented the following method for two practical reasons. LCD and its variants LCDa, LCDb and LCDc require as input one or more instrumental variables. For a fair comparison between BLCD and LCD (and its variants) we have incorporated the use of the same instrumental variable(s) as prior knowledge. When BLCD is provided with prior knowledge, we refer to the variant of the algorithm as BLCDpk. For simulated data from synthetic or known networks, we have used the root nodes of the generating structure as the instrumental variables. Likewise, for the real-world dataset that

we used it was not feasible to have our domain experts assign priors to more than three thousand pairwise relationships in the dataset (see Section 6.4.1).

Assume that we can identify one or more variables in a domain as root nodes. This prior knowledge could come from a domain expert or from published literature. When a tetraset W_1, W_2, X, Y is being evaluated, recall that X is fixed[1] and hence we can generate three "Y" patterns. If prior knowledge informs that X is a root node, that tetraset will not be evaluated. The probability of all the three "Y" patterns will be assigned 0. If Y is a root node, one "Y" pattern is assigned a probability 0. When W_1 and/or W_2 is a root node, when we sum all 543 models only models satisfying the root constraint are scored and summed when $P(X \rightarrow Y|D)$ is computed using Equation 5.1. Figure 30 illustrates the effect of making Y a root node using prior knowledge, and Figure 31 illustrates the effect of making W_1 and W_2 root nodes using prior knowledge.

[1] W_1, W_2 and Y are in the Markov blanket of X.

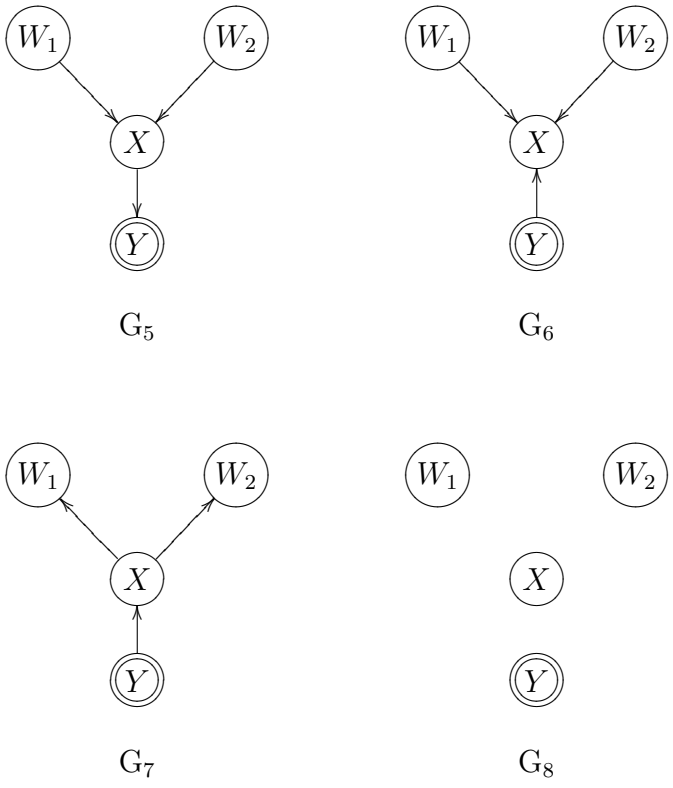


Figure 30: Prior knowledge (Y is a root node) applied to the models from Figure 25. $P(G_5)$ will be assigned a value of 0.

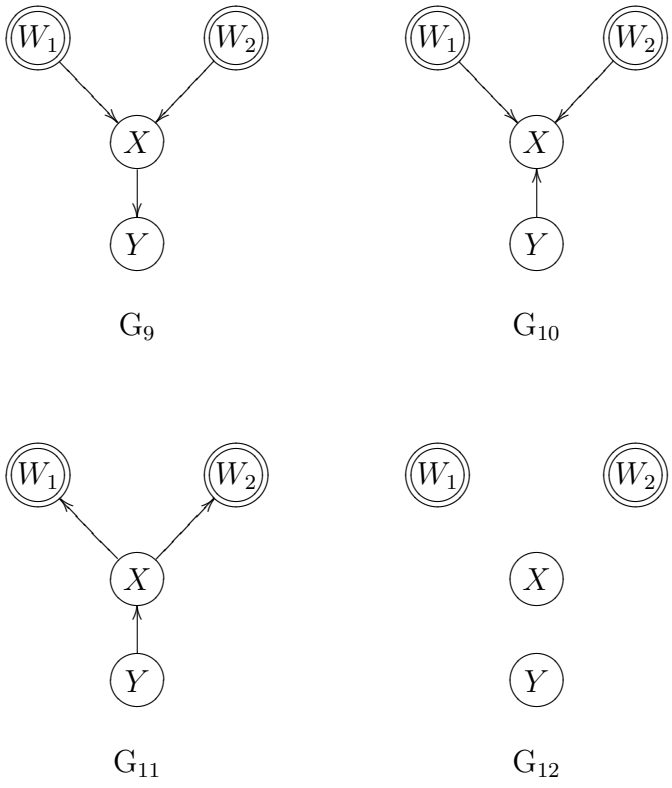


Figure 31: Prior knowledge (W_1 and W_2 are root nodes) applied to the models from Figure 25. $P(G_{11})$ will be assigned a value of 0.

The pseudocode for the procedure used in BLCD for incorporating prior knowledge is given below: Instead of Step 4 of BLCD (see Section 5.1.4), call Procedure PriorKB.

Procedure PriorKB (Additional input $\mathbf{R}$ set which is the set of instrumental variables)
DO
PK4: If $X \in \mathbf{R}$,
Foreach "Y" structure
DO
$P(X \rightarrow Y|D) = 0$;
OD
ELSE
DO
Foreach "Y" structure
DO
If ($Y \in \mathbf{R}$)
$P(X \rightarrow Y|D) = 0$; (see Figure 30)
ELSE
DO
If ($W_1 \in \mathbf{R}$ or $W_2 \in \mathbf{R}$)
DO
Score only structures that do not violate this constraint. (see Figure 31)
Derive $P(X \rightarrow Y|D)$ using Equation 5.1.
OD
OD
OD
OD
OD

Other types of prior knowledge such as *terminal nodes* could be similarly incorporated into this framework. If a node $X \in \mathbf{V}$ does not causally influence any of the other nodes in

V, it is called a terminal node. The leaf nodes of a CBN are terminal nodes. One or more terminal nodes could be provided as prior knowledge to BLCD.

5.2 EXTENSIONS TO BLCD

In this section we explore two extensions to the BLCD framework.

5.2.1 BLCDvss: Making use of shielded colliders

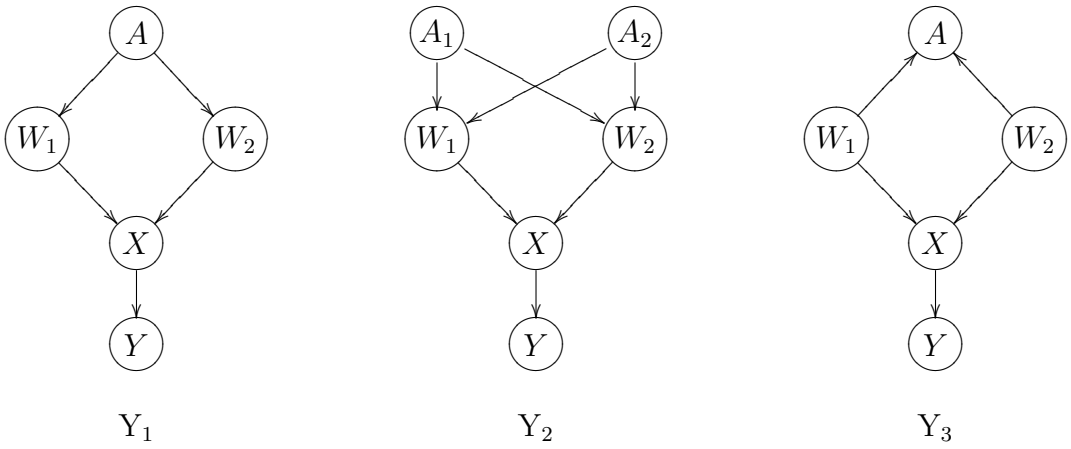


Figure 32: Two Mshielded "Y" structures ("Y_1" and "Y_2") and one unshielded "Y" structure ("Y_3").

Recall that BLCD requires that a node X be an unshielded collider in order to discover the causal effects of X. In the extended BLCD framework we relax that assumption and search for conditionally unshielded colliders, where W_1 and W_2 are dependent due to measured variables. This measured variable is denoted by A, and if more than one by $\mathbf{A}$. We also refer to these shielded colliders as conditionally unshielded colliders because conditioning on the measured variable makes them unshielded. When a conditionally unshielded collider X forms a "Y" structure with an effect variable Y, we call the resulting structural configuration an Mshielded (or conditionally unshielded) "Y" structure. Figure 32 shows two such Mshielded "Y" structures ("Y_1" and "Y_2"). Figure 32 also shows an unshielded "Y" structure ("Y_3") and this type of structure will not be conditioned by BLCDvss[2] (see Step 6.2 of Procedure ApplyVshield given below). By conditioning on the variable A or the

[2]These types of structures will be evaluated by BLCD.

set **A**, the shielded "Y" structure can be converted to the traditional "Y" and $P(X \rightarrow Y|D)$ computed using Equation 5.1. Note that the current implementation of BLCDvss only conditions on one A variable. Hence if "Y_2" is the generating structure, we will not be able to infer $P(X \rightarrow Y \mid D)$ as conditioning on A_1 or A_2 does not make $W_1 \perp\!\!\!\perp W_2$. The pseudocode for BLCDvss is provided below.

Additional steps for BLCDvss.

6. If $P(X \rightarrow Y \mid D) \leq t$, then

    Procedure ApplyVshield.

Procedure ApplyVshield

DO

6.1 Derive the set of nodes that are dependent on W_1. Label the set $\mathbf{S}_1$.

Derive the set of nodes that are dependent on W_2. Label the set $\mathbf{S}_2$.

Obtain the intersection of sets $\mathbf{S}_1$ and $\mathbf{S}_2$. Label the new set $\mathbf{S}_3$.

$\mathbf{S}_4 = \mathbf{S}_3 \setminus \{X, Y\}$.

6.2 Foreach node $A \in \mathbf{S}_4$

/* Identify node A which if conditioned on makes $W_1 \perp\!\!\!\perp W_2$ and. */

/* Exclude node A if A is a child of W_1 and W_2 ($W_1 \longrightarrow A \longleftarrow W_2$)[2]. */

If $(W_1 \perp\!\!\!\perp W_2 \mid A)$

Return A.

6.3 Derive $P(X \rightarrow Y|D, A)$ using Equation 5.1.

6.4 GOTO Step 5 of BLCD.

OD

[2]This step will exclude node A if it is a child of W_1 and W_2, as $W_1 \not\perp\!\!\!\perp W_2 \mid A$

Another approach pursued in extending BLCD was to look for additional causal influences in data by constructive induction. The following section explores this issue.

5.2.2 BLCDcv: Combining X and Z variables

This section introduces the concept of combining two variables. We can use constructive induction to combine two variables A and B to form an aggregate variable C by taking the cartesian product of A and B, which is represented as $\{A \times B\}$. Consider five discrete random

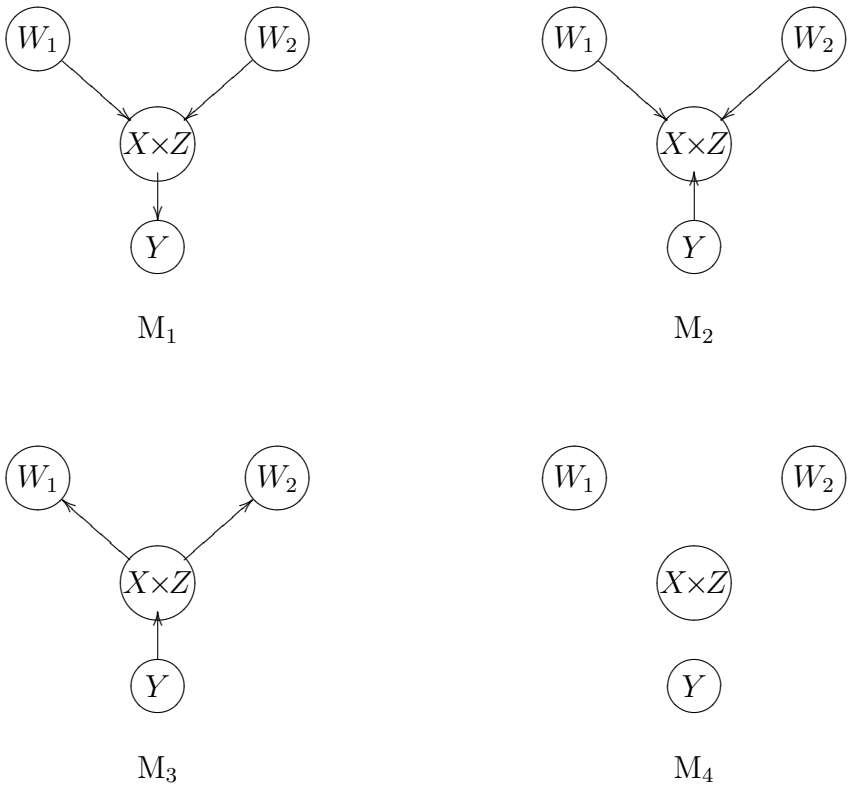


Figure 33: Causal models for bounding the causal effect of X on Y. $X \times Z$ denotes the cartesian product of X and Z.

observed variables—W_1, W_2, X, Y, and Z, where Z is any node in the Markov blanket of Y (excluding W_1, W_2, and X) that is also probabilistically dependent on Y. Figure 33 shows four models out of the total possible 543 when X and Z are combined and represented as a single variable $\{X \times Z\}$. The $P(\{X \times Z\} \rightarrow Y \mid D)$ can now be computed using Equation 5.1. The advantage of doing so is that X and Z together might be more predictive of Y than either alone. The disadvantage is that when BLCDcv outputs that $\{X \times Z\} \rightarrow Y$, any one of the following interpretations can hold:

1. Both X and Z causally influence Y.
2. X causally influences Y, and Z acts as a non-causal covariate.
3. Z causally influences Y, and X acts as a non-causal covariate.

BLCDcv currently does not attempt to differentiate the above causal hypotheses.

The additional steps for BLCDcv are given below:

Additional steps for BLCDcv.

Procedure CombVar

DO

$\mathbf{Z} = \text{MB}(Y) \setminus \{W_1, W_2, X\}$.

Foreach $Z \in \mathbf{Z}$

DO

7.1 Create $\{X \times Z\}$.

7.2 Derive $\text{P}(\{X \times Z\} \rightarrow Y \mid D)$ using Equation 5.1.

7.3 GOTO Step 5 of BLCD.

OD

OD

6.0 EXPERIMENTAL METHODS

In this Chapter we describe the experiments used to test the dissertation hypothesis.

Evaluation of discovered causal postulates is more difficult in comparison to assessments of machine learning algorithms for classification tasks or for association rule mining (Cooper, 1999, page 55). As gold standard labels are available, assessing the accuracy of classifiers and comparing the performance of various classification algorithms is clearly feasible, if not always completely straightforward. For association rules, grading can be done based on the statistical dependence properties of the attributes represented in the rules.

Evaluation of causal output can be done using:

1. Synthetic (artificial) causal models.
2. Expert designed causal models.
3. Real-world databases and some assumed gold standard for judging causal relationships.

For items 1 and 2, the generating causal model is known and can act as a gold standard. The parameters are also explicitly defined in terms of marginal and conditional probabilities. For 3, the data generating causal process is often not completely known. However, some insight into the causal mechanisms can be found in domain literature and from subject experts.

We evaluated the dissertation *hypothesis* (see Section 1.2) by comparing the performance of the different causal discovery algorithms in context. We compared BLCD, BLCD variants, LCD variants (LCDa, LCDb, LCDc), PC, FCI, and OR[1]. Specifically, the following pairwise comparisons were used in the hypothesis evaluation, as all these algorithms output direct

[1]BLCD and its variants are described in Sections 5.1, 5.2 and LCDa, LCDb, and LCDc in Section 4.1.1. PC is described in detail in Section 3.1.1, FCI in Section 3.1.2 and OR in Section 3.3

causal relationships[2]:

1. BLCD vs PC and FCI, not assuming instrumental variables.
2. BLCD vs OR.
3. OR vs PC and FCI.

These comparisons involved the use of both real-world data and synthetic data generated from representative known causal Bayesian networks. We implemented BLCD and its variants (BLCDpk, BLCDvss, BLCDcv), LCDa, LCDb, and LCDc in the C programming language. We obtained the Tetrad program that implements the PC and FCI algorithms from Dr. Peter Spirtes. Specifically the Tetrad version tetradcmd-4.3.3-1 was used for all of our experiments. The OR implementation was obtained from Professor Andrew Moore of Carnegie Mellon University (http://www.autonlab.org/autonweb/software/10530.html).

In the evaluation of causal output, we can consider two dimensions: (1) qualitative or structural and (2) quantitative or parameterization. The focus of our evaluation was qualitative or structural because the parameters can be estimated from data once the structure is known, particularly for unconfounded causal relationships. For confounded causal relationships, if the confounding is by measured variables, the parameters can be estimated by conditioning on the measured confounders. These relationships are described in Section 2.3. The output of the algorithm was compared with the data generating structure in the case of expert designed networks and scored as explained in Section 6.3.2. For real-world databases, evaluation was more exploratory in nature and was based on the subjective assessment of two domain experts.

6.1 Y STRUCTURES FROM PC, FCI AND OR

In this section we describe how the output of PC, FCI and OR was processed in order to map their output to Y structures. Since these algorithms output a global model, the mapping was done to all Y structures, both confounded (by measured variables only) and unconfounded.

[2]If there is an arc from variable X to variable Y in a given model, then the variable X is said to be a direct cause of variable Y in that model.

We refer to these Y structures as global Y structures (Figure 34 shows three different Y structures). Since the FCI can model hidden variables and its output has a clear causal semantics, the directed arcs that were output by FCI were mapped to Y structures in the generating networks.

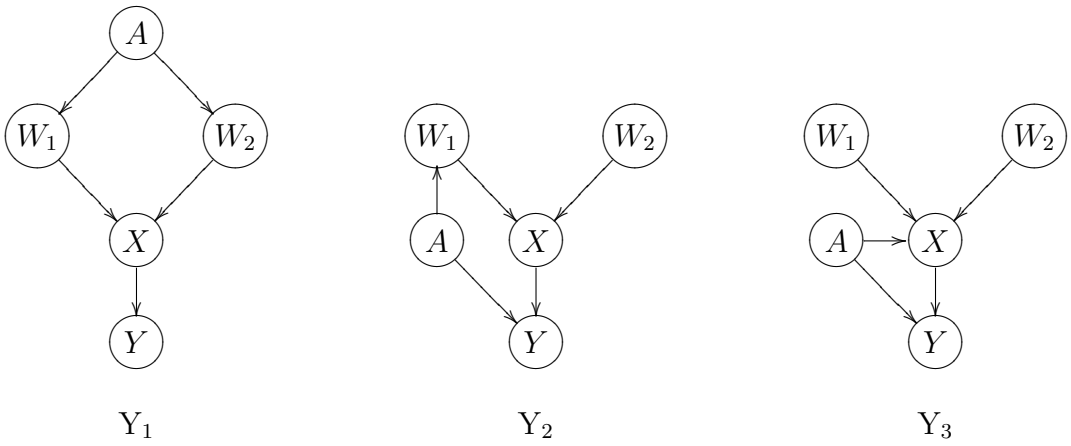


Figure 34: An Mshielded "Y" structure ("Y_1") and two confounded "Y" structures ("Y_2" and "Y_3").

The PC algorithm assumes causal sufficiency and outputs both directed and undirected arcs. A post-processing step is performed on the set of arcs to identify the global Y structures. Note that the OR algorithm outputs a global Bayesian network consisting of all the variables of the input dataset. A post-processing step is performed on the output (Bayesian network) to identify the global Y structures. For PC and OR these Y structures were mapped to the global Y structures of the data generating networks. In short we get the intersection set of Y structures present in the output and the generating model.

6.2 SYNTHETIC BAYESIAN NETWORKS

Aliferis and Cooper evaluated the K2 algorithm by constructing artificial BNs and creating simulated databases from them (Aliferis & Cooper, 1994). As the causal network generating

the data is known, comparing the output of a causal discovery algorithm against the gold-standard is possible. But there are limitations to this approach (Cooper, 1999, page 54):

1. What is being evaluated is the internal validity of discovery methods with reference to an assumed BN model of causality. External validity with respect to the real-world causal discovery is not being addressed in such a scenario.
2. Parameterization of synthetic networks is problematic. A set of probabilities have to be assigned for obtaining simulated datasets from synthetic BNs, and assessing these probabilities can be difficult.
3. Assessing structure can also be difficult. For example, if the algorithm recovers the generating structure, we can only surmise that the method will succeed in the real world if the synthetic structure mimics the real-world causal mechanisms.

   Because of the above-mentioned drawbacks, we did not use artificial BNs for evaluation.

6.3 EXPERT DESIGNED CAUSAL NETWORKS

Domain experts can construct CBNs based on their expertise and knowledge from literature. They must assess both the structure and the underlying probability parameters. Alarm, Pathfinder and Munin are expert-constructed BNs in the domain of medicine. These networks are likely to reflect at least to some degree the structure and parameters of causal phenomena occurring in the real world. But generating these causal models is time consuming in general and constrained by the expert's understanding of the causal mechanisms (Cooper, 1999, page 55). We used these expert-crafted networks for evaluation as described in Section 6.3.1. We first discuss a general approach to evaluation for such networks and move on to a description of the individual networks (Sections 6.3.2– 6.3.6). In Section 6.3.8 we discuss the evaluation measures that were used.

6.3.1 Network evaluation

We first describe how the node pairs and arcs of a causal Bayesian network can be categorized.

Each node pair (X, Y) in a causal Bayesian network is categorized as follows (see Section 2.3 for definitions):

1. Causal and unconfounded pairs (CUP).
2. Causal and confounded pairs (CCP).
3. Confounded-only pairs (COP).
4. Independent pairs (IP).

Table 7: Categories of node pairs in the Alarm, Hailfinder, Barley, Pathfinder, and Munin networks

Category	Alarm	Hailfinder	Barley	Pathfinder	Munin
CUP	167 (25%)	307 (20%)	206 (18%)	109 (1.9%)	1785 (10%)
CCP	56 (8%)	89 (6%)	333 (30%)	148 (2.5%)	561 (3.2%)
COP	78 (12%)	236 (15%)	169 (15%)	5629 (95.6%)	7430 (41.8%)
IP	365 (55%)	908 (59%)	420 (37%)	0 (0%)	7990 (45%)
Total	666 (100%)	1540 (100%)	1128 (100%)	5886 (100%)	17766 (100%)

CUP–Causal and unconfounded pairs, CCP–Causal and confounded pairs, COP–Confounded only pairs, IP–Independent pairs

Table 7 gives the distribution of these different categories for the various networks. When categorizing a pair (X, Y) as *causal and unconfounded* or *causal and confounded*, the direction of the path between X and Y is important. If the direction is incorrect, two types of mis-categorization can occur: *causal and unconfounded reversed* and *causal and confounded reversed*. In all the networks except Pathfinder, CUP is at least 10% of the total pairs. Likewise, the percentage of COP is less than 50% for all the networks except Pathfinder. Note that in the Pathfinder network 95.6% are of COP category. This pattern is explained by the fact that the disease node (Fault) is the parent of 105 out of a total of 131 nodes in the network. The Pathfinder network does not have any independent pairs.

Note that causal pairs such as $A1, A4$ or $A1, A5$ (see Figure 35) will not be output by PC, FCI and BLCD algorithms (when they operate ideally, as designed) as there are no arcs

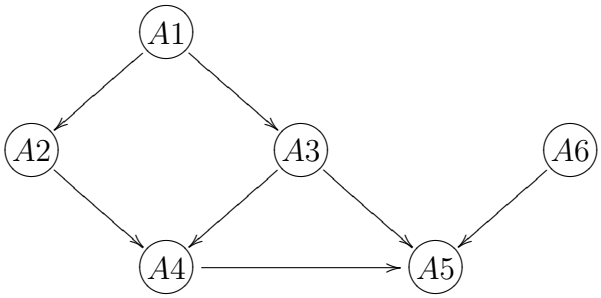


Figure 35: A causal Bayesian network structure with six nodes and seven arcs.

connecting these pairs directly. These algorithms search for direct causal influences. On the other hand LCD and its variants are designed to output these pairs as causal. Hence LCD and its variants were evaluated based on pair categories and the other algorithms (OR, PC, FCI, BLCD and BLCD variants) that output direct causal relationships were evaluated based on Y structures.

Table 8: Nodes and arcs in the Alarm, Hailfinder, Barley, Pathfinder, and Munin networks

Category	Alarm	HF	Barley	PF	Munin
Nodes	37	56	48	131	189
Root nodes	12	17	10	1	34
Arcs	46	66	84	195	282
Causal and unconfounded arcs (CUA)	37	54	33	107	238
Causal and confounded arcs (CCA)	9	12	51	88	44

HF: Hailfinder; PF: Pathfinder

Table 8 gives the distribution of the different categories based on arcs for the various networks. The table additionally provides a count of arcs that are unconfounded and arcs that are confounded. Recall that there are no confounded-only or independent arcs (see Section 2.3). Note also that all causal and unconfounded arcs (CUA) are causal and unconfounded pairs (CUP) but not vice versa. With reference to Figure 35, $A1, A2$ and $A1, A3$

are CUA as well as CUP, while $A1, A5$ is a CUP but not a CUA. Similar property holds for CCA and CCP also.

We now describe a modified framework for computing precision and recall for the various algorithms based on their causal discovery approach. LCD outputs causal and unconfounded pairs (causal and unconfounded pair of nodes connected by one or more arcs), BLCD outputs causal and unconfounded arcs, and PC and FCI output both confounded (by measured variables) and unconfounded arcs. As an example, Table 9 enumerates the ideal output of LCD, BLCD, PC, and FCI assuming the generating causal structure given in Figure 36.

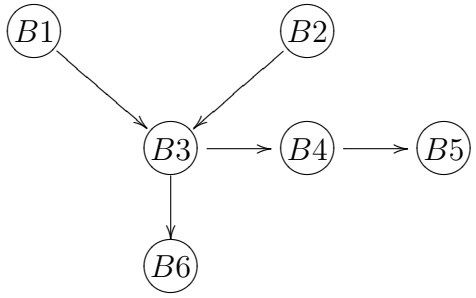


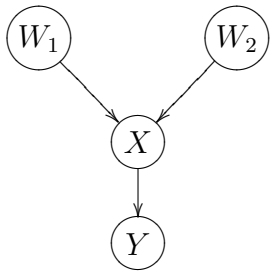
Figure 36: A causal Bayesian network structure with six nodes and five arcs.

Table 9: A best case scenario output for PC, FCI, BLCD, and LCD algorithms for the causal Bayesian network shown in Figure 36.

Algorithm	Arcs output
PC	B1 → B3; B2 → B3; B3 → B4; B4 → B5; B3 → B6
FCI	B3 → B4; B4 → B5; B3 → B6
BLCD	B3 → B4; B3 → B6
LCD	B3 → B6; B3 → B4; B3 → B5; B4 → B5

The output of LCD algorithms will be graded based on the total number of *causal and unconfounded pairs* (CUP) in the generating structure. For example, for computing recall values for the Alarm network, the denominator will be 167 for LCD (see row 1, Table 7).

LCD algorithms will also be graded based on *discoverable causal and unconfounded pairs* (DCUP). Recall that LCD requires W variables for causal discovery and LCD algorithms cannot discover causal and unconfounded pairs W, Y. For each network dataset, the root nodes of the generating network will be designated as the W variables. Hence DCUP are a subset of CUP such that there are no root nodes of the generating network represented in DCUP.



G_1

Figure 37: A Y structure. $X \to Y$ is a Y arc (YA).

BLCD, BLCD variants, PC, FCI and OR will be graded based on Y arcs (YA) output by the algorithm. An YA is the arc that forms the tail of a Y structure. For example, if W_1, W_2, X, and Y form an Y structure (see Figure 37), the $X \to Y$ arc is referred to as the YA. As PC, FCI and OR output global models, we include Y arcs that are part of unconfounded and of measured confounded Y structures (see Figure 34)[3]. The union of all Y arcs (both confounded and unconfounded) is referred to as GYA (global Y arcs). Note that GYA also represent what is causally discoverable theoretically (assuming observational data and no prior knowledge) across all algorithms that output direct causal relationships (BLCD and its variants, PC, FCI and OR). Recall that OR outputs a global Bayesian network

[3]If the confounders of a YA are all measured variables, by conditioning on them we obtain an unconfounded YA.

and does not claim a causal interpretation for the arcs that are output. PC outputs both directed and undirected arcs while also assuming causal sufficiency. Hence post-processing of the output is done to identify the GYA patterns of OR and PC and then compared with the GYA of the data generating network. FCI does not assume causal sufficiency and when it outputs an arc, it represents a causal influence. The arcs output by FCI will be compared with GYA of the data generating network.

For an algorithm such as BLCD, an unconfounded Y structure is needed in the data generating model (DGM) to establish causality from observational data. For algorithms that output a global network, both confounded and unconfounded Y structures in the DGM can establish causality, as the confounded Y structures can sometimes be rendered unconfounded by conditioning on the confounding variables, when they are measured.

6.3.2 The Alarm network

The Alarm causal Bayesian network contains 37 nodes and 46 causal arcs. Each node can have two to four possible states. Dr. Ingo Beinlich developed Alarm to model potential anesthesia-related problems in the operating room (Beinlich et al., 1990). His experience as an anesthesiologist and medical knowledge from the literature went into the development of the Alarm network. Alarm has been used extensively in evaluations of Bayesian network induction. We believe it remains a useful standard benchmark.

It has 37 nodes, 46 arcs, and the total number of possible distinct pairs is 666.

6.3.3 The Hailfinder network

Hailfinder (Abramson et al., 1996; Edwards, 1998) was designed to predict severe summer storms in Colorado. It has 56 nodes, 66 arcs, and the total number of distinct pairs is 1540.

6.3.4 The Barley network

Barley was created as a decision support tool for barley cultivation (Kristensen & Rasmussen, 2002). Barley has 48 nodes, 84 arcs, and the total number of distinct pairs is 1128.

6.3.5 The Pathfinder network

The Pathfinder Bayesian network was constructed by Heckerman and Horvitz and a domain expert. It represents the domain of lymph node pathology with 130 feature nodes and the class node (disease pathology) representing 62 conditions, both benign and malignant (Heckerman et al., 1992; Heckerman & Nathwani, 1992). It has a total of 131 nodes, 195 arcs, and the total number of distinct node pairs is 5886. The Pathfinder system was designed to assist pathologists in the differential diagnosis of lymph node diseases. It is not clear from the system description if the network was built with a causal semantics. However, an examination of the directionality of the edges indicates that most of the edges have a causal semantics (particularly the ones going from the disease node to the feature nodes).

6.3.6 The MUNIN network

MUNIN (Andreassen et al., 1987; Suojanen et al., 1997) is a probabilistic medical decision support system for the domain of peripheral nerve and muscle disorders. It has evolved over the years, starting in the late eighties. The MUNIN version that was used has 189 nodes, 282 arcs, and the total number of distinct node pairs is 17766. It encodes the causal relationships among the variables probabilistically in a causal probabilistic network. Most of the relationships appear causal on inspection of the network. This is a large network, and it is useful to test the performance of the various algorithms on a network dataset of this scale.

6.3.7 Dataset generation

For causal discovery we generated simulated training instances by logic sampling (Henrion, 1986). We varied the number of training instances generated and used for various experiments. The same set of training cases were given as input to all the discovery algorithms.

6.3.8 Evaluation metrics for simulated causal network data

For the expert designed causal networks, the DGM is known and hence causal discovery performance can be assessed as a function of the true positives, false positives, true negatives, and false negatives. Metrics such as *precision* and *recall* were computed using Equations 1.1 and 1.2.

Using various thresholds of probability for scoring a relationship, the metrics of precision and recall were plotted for the different networks and algorithms.

6.4 REAL-WORLD DATABASES

Evaluation of causal output from real-world observational databases is challenging. A gold-standard evaluation would involve experimental verification of a postulated causal hypothesis by manipulating the variable postulated to be causal and measuring the effect under controlled conditions. However, this approach was not feasible for me to pursue. If the causal mechanisms are already known with confidence (for example, published in peer-reviewed, high quality journals), they can be used as gold standards. Domain experts can also evaluate the causal output based on their causal knowledge of the field. However, novel relationships output by causal discovery algorithms are not easily amenable to expert interpretation and validation.

Section 6.4.1 describes the large real-world dataset that I used to evaluate causal discovery algorithms. Section 6.4.2 discusses the prior knowledge that was input for LCDa, LCDb, LCDc and BLCDpk.

6.4.1 Infant Birth and Death Dataset

I used the U.S. Linked Birth/Infant Death dataset for 1991 (National Center for Health Statistics, 1996). This dataset consists of information on all the live births in the United States for the year 1991. It also has linked data for infants who died within one year of birth. More than two hundred variables containing various maternal, paternal, fetal and

infant parameters were available. For the infants who died within the first year, additional data on mortality, including cause of death, is reported. The records total more than four million and the infant death record number is 35,496. I have selected a total of 87 variables after eliminating redundant variables and variables not of clinical interest, such as ID number. The 87 variables constitute 3741 unique pairwise relationships that fall into one of the four possible mutually exclusive expert-rated categories: $X \rightarrow Y$ causal effect, $Y \rightarrow X$ causal effect, $X <> Y$ (X independent Y or X and Y confounded by a common parent/ancestor), and finally *undetermined*. In other words, the rating $X <> Y$ means that the nodes X and Y are not causally related to each other. A pairwise relationship would be *undetermined* if the domain experts were not able to categorize it into one of the above three categories. As it may not be feasible to categorize these thousands of relationships from literature or from experts for purposes of a gold standard, we evaluated the purported causal relationships that were output by the discovery algorithms (which are a small subset of the total number of possible pairs) using domain experts to judge them. Thus, we can estimate precision but not the standard recall exactly. A relative version of recall however, can be obtained as follows. If $\mathcal{T}$ is the total number of relationships determined as causal by the domain experts over all the rated output of all the algorithms, we take $\mathcal{T}$ as representing the known causal relationships in deriving a relative recall.

Two experts who graded the output are practicing neonatologists trained in newborn medicine and working in teaching hospitals in the US. The experts were provided with the union of all the relationships output as causal by the various algorithms. A total of 252 relationships (along with an explanation of the variables as provided in the dataset documentation) were made available to each of the experts. They were asked to grade a pair X, Y as follows:

1. C if $X \rightarrow Y$.
2. CR if $Y \rightarrow X$.
3. N if $X <> Y$ (not causally related).
4. U if unable to categorize as one of the above.

The following procedure was used to create the gold standard category using the labels

of the two experts.

Procedure GoldStandard

DO

Foreach output pair X, Y

DO

If the grade of expert-one is C or the grade of expert-two is C

The gold-standard grade = C;

Else If the grade of expert-one is CR or the grade of expert-two is CR

The gold-standard grade = CR;

Else If the grade of expert-one is N or the grade of expert-two is N

The gold-standard grade = N;

Else the gold-standard grade = U;

OD

OD

A summary of the gold standard labels are given in Table 10.

Table 10: Gold standard labels for Infant data

Gold standard label	Number	Percent
C ($X \rightarrow Y$)	75	29.8%
CR ($Y \rightarrow X$)	96	38.1%
N ($X <> Y$)	79	31.3%
U (unknown)	2	0.8%

$X <> Y$: Not causally related.

We also created a platinum standard category from the labels of the two experts using the following conservative procedure.

Procedure PlatinumStandard

DO

Foreach output pair X, Y

DO

If the grade of expert-one is C and the grade of expert-two is C

The platinum-standard grade = C;

Else If the grade of expert-one is CR and the grade of expert-two is CR

The platinum-standard grade = CR;

Else If the grade of expert-one is N and the grade of expert-two is N

The platinum-standard grade = N;

Else the platinum-standard grade = U;

OD

OD

A summary of the platinum standard labels are given in Table 11.

Table 11: Platinum standard labels for Infant data

Gold standard label	Number	Percent
C ($X \rightarrow Y$)	28	11.1%
CR ($Y \rightarrow X$)	56	22.2%
N ($X <> Y$)	57	22.6%
U (unknown)	111	44.1%

$X <> Y$: Not causally related.

6.4.2 Prior knowledge for real-world datasets

The instrumental variables were selected from domain knowledge available from literature. In particular, I used child gender and maternal race for the instrumental variables. The union of all the relationships output by the different algorithms were categorized by the experts. The different algorithms were compared using these categorizations.

6.5 EXPERIMENTAL RUNS

For comparing the performance of the algorithms LCDa, LCDb, LCDc, BLCD (and BLCD variants), PC, FCI, and OR, we varied the size of the dataset for both simulated data and real-world data (Infant Birth Death data). The number of instances for these experimental runs varied from 100 to 20,000 for the simulated datasets. For the Infant dataset, the sample sizes were 10,000, 20,000, 40,000, and 100,000.

For the Alarm, Hailfinder, Barley, Pathfinder and Munin network datasets, we compared the output of the algorithms with gold-standard data-generating networks as discussed in Section 6.3.1 and computed structural metrics as described in Section 6.3.8. For the Infant Birth Death dataset we compared the different causal discovery algorithms as discussed in Section 6.4.1.

PC and FCI algorithms were run with the default options using a significance level (p value) of 0.05 [4]. OR was also run with default options. If these algorithms went out of memory with certain datasets, the default options were modified in such a way so as to facilitate experimental output. For example, with PC and FCI lowering the p value to 0.01 and with OR, lowering the number of candidate parents of a node from eight (default) to 4,2 or 1 was done. LCDa, LCDb, and LCDc were run with a 0.9 threshold for the Bayesian dependence and independence tests. For BLCD and BLCD variants, we used an upper limit of ten nodes when deriving the Markov blanket of a node. The default threshold of 0.5 was used with simulated datasets to output a relationship as causal. A lower threshold of 0.1

[4]We also tried running PC and FCI with significance levels greater than 0.05. PC could be run on Alarm dataset with higher significance levels, but FCI ran out of memory.

was used with Infant data. Using the default threshold of 0.5 with Infant data resulted in either no relationships being output or only a very small number being output.

Two machines M_1 and M_2 were used to run all the experiments. M_1 is a PC with a single 3 GHz Intel processor, 2 GB RAM and runs the Linux operating system. M_2 is a PC with two 3 GHz Intel processors, 6 GB RAM and runs the Linux operating system. The machine M_1 was used for Alarm, Hailfinder, Barley and Infant datasets, while machine M_2 was used for the Pathfinder and Munin datasets. We also recorded the runtimes for all the experiments.

7.0 RESULTS

In this Chapter we present the results of the experimental runs described in Chapter 6. Table 12 summarizes the algorithms that output both direct and indirect causal relationships and Table 13 lists the algorithms that output direct causal relationships only. All the algorithms enumerated in Tables 12 and 13 were used to test the dissertation hypothesis. Figures 38 and 39 list some representative examples of the different types of Y structures present in the data generating networks that we will refer to from time to time. Table 14 matches the Y structures from Figures 38 and 39 to the algorithms from Table 13 that output them. Section 7.1 presents the results related to the OR algorithm, Section 7.2 gives the results pertaining to the PC and FCI algorithms, Section 7.3 provides results based on BLCD, BLCDpk, BLCDcv and BLCDvss algorithms, and Section 7.4 presents the results of the LCD class of algorithms. Finally, Section 7.5 gives the results based on the Infant dataset. Additional results are also provided in Appendix A.

The results are based on what is theoretically discoverable for each of the algorithms. For the algorithms that discover direct causal relationships (OR, PC, FCI and the BLCD algorithms) we also present results based on the union of what is theoretically discoverable over all these algorithms. The union of what is theoretically possible to discover is the set of all the arcs (causal and unconfounded as well as causal and Mconfounded[1]) represented in the Y structures of the generating networks. The results presented below are based on sample sizes that vary between 100 and 100,000 instances.

Tables 7, 8, and 15 contain summaries of the properties of the different networks used in this study, and Section 6.3.1 describes how the different expert-crafted networks were

[1]By Mconfounded here I mean *measured* confounders that are therefore in the datasets made available to the algorithms.

Table 12: A synopsis of the LCD algorithms used

Algorithm	Description	Extra Knowledge	Output
LCDa	Local causal discovery algorithm that is based on constraint-based dependence and independence tests. Requires just one positive independence test.	Assumed root nodes[1]	Unconfounded causal relationships
LCDb	Local causal discovery algorithm that is based on constraint-based dependence and independence tests. Requires positive independence tests with two root nodes.	Assumed root nodes	Unconfounded causal relationships
LCDc	Local causal discovery algorithm that is based on constraint-based dependence and independence tests. Requires positive independence tests with all root nodes.	Assumed root nodes	Unconfounded causal relationships

[1] These are nodes that are not caused by any of the other measured variables in the dataset.

used for evaluation. The reader is referred to Section 2.3 for definitions of causal influence, confounded and unconfounded causal relationships and for a description of the terms CUP (Causal and unconfounded pairs), CCP (Causal and confounded pairs), COP (Confounded only pairs), IP (Independent pairs), CUA (Causal and unconfounded arcs), and CCA (Causal and confounded arcs).

Table 13: A synopsis of BLCD, OR, PC and FCI

Algorithm	Description	Extra Knowledge	Output
BLCD	Bayesian local causal discovery algorithm based on a Bayesian scoring function to evaluate the models given the data.	None	UDCR[1]
BLCDpk	Variant of BLCD that uses prior knowledge in the form of root nodes.	Assumed root nodes	UDCR
BLCDvss	Variant of BLCD that converts "Mshielded" (see Section 5.1.1 for a definition) into "unshielded" colliders in discovering additional causal influences.	None	UDCR
BLCDcv	Variant of BLCD that combines variables X and Z, and then searches for the effects of this combined variable $X \times Z$.	None	UDCR (see also Section 5.2.2)
PC	PC algorithm implemented in the Tetrad package.	None	Mconfounded[2] and UDCR
FCI	FCI algorithm implemented in the Tetrad package.	None	Mconfounded and UDCR
OR	Optimal Reinsertion algorithm.	None	Mconfounded and UDCR

UDCR: Unconfounded direct causal relationships

[1] If there is an arc from variable X to variable Y in the generating network, X is a direct cause of Y.

[2] Confounded by measured variables that are present in the datasets made available to the algorithms.

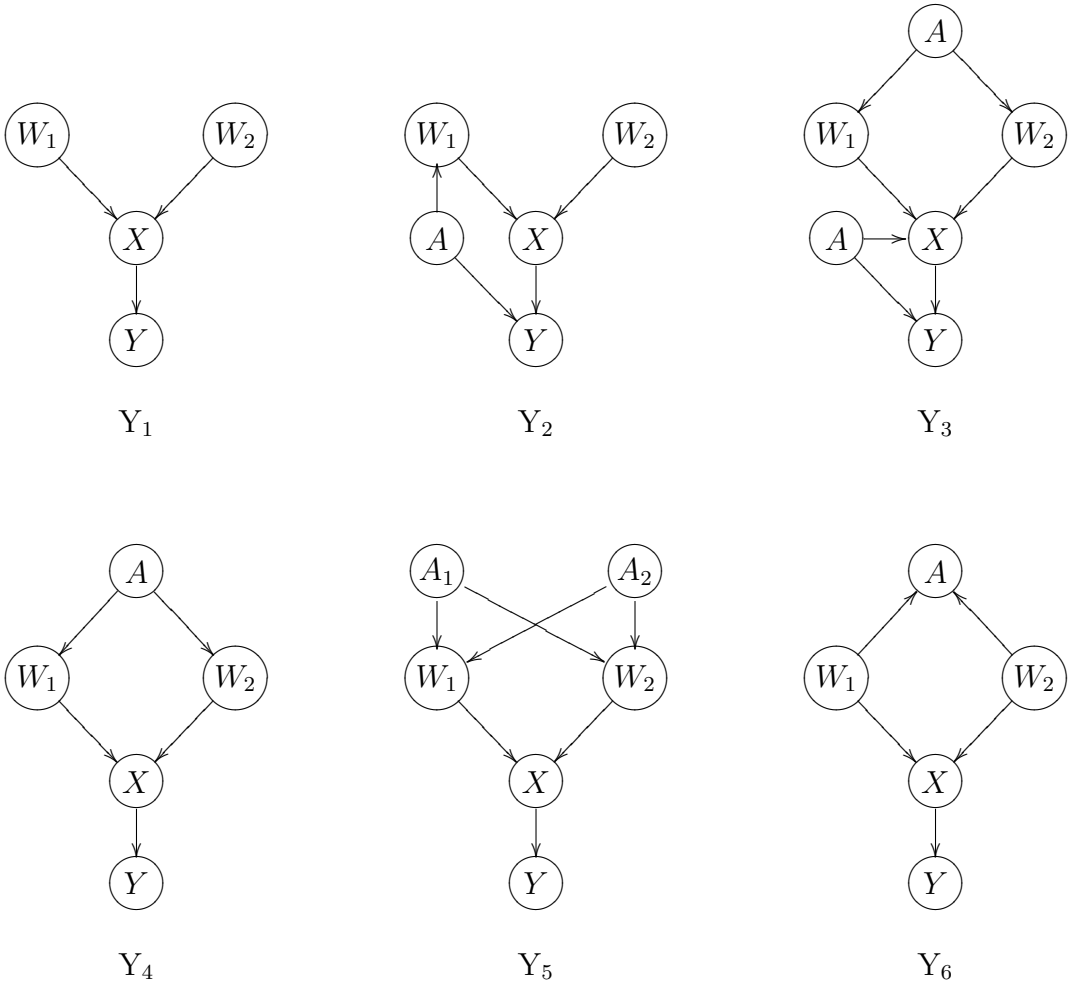


Figure 38: Six "Y" structures.

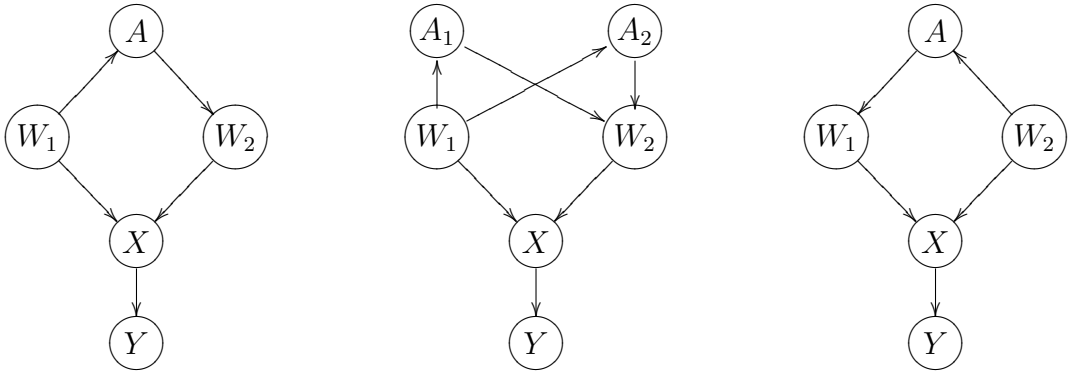


Figure 39: Three additional "Y" structures.

Table 14: Types of Y structures and algorithms that output them (see also Figures 38 and 39)

Y structure	Category	Algorithms
Y1	GY, UcUsY	OR[1], PC[1], FCI, BLCD, BLCDpk, BLCDvss, BLCDcv
Y2	GY, CUsY	OR, PC, FCI
Y3	GY, CMsY	OR, PC, FCI
Y4	GY, UcMsY	OR, PC, FCI, BLCDvss
Y5	GY, UcMsY	OR, PC, FCI, BLCDvss[2]
Y6	GY, UcUsY	OR, PC, FCI, BLCD, BLCDpk, BLCDvss, BLCDcv
Y7	GY, UcMsY	OR, PC, FCI, BLCDvss
Y8	GY, UcMsY	OR, PC, FCI, BLCDvss[2]
Y9	GY, UcMsY	OR, PC, FCI, BLCDvss

[1] OR and PC output is post-processed for identifying the GY structures.

[2] In theory BLCDvss can output these Y structures by conditioning on A_1 and A_2. However, the current implementation of BLCDvss supports conditioning on only one such variable at a time.

GY: Global Y

C: Confounded

Uc: Unconfounded

Us: Unshielded

Ms: Mshielded

Table 15: Types of "Y" structures in the Alarm, Hailfinder, Barley, Pathfinder, and Munin networks

Category	Alarm	HF	Barley	PF	Munin
Unshielded and unconfounded"Y" (UsUcY)	6	11	2	0	96
Mshielded and unconfounded "Y" (MsUcY)	1	2	1	0	16
Union of UsUcY and MsUcY	6	13	3	0	112
Global "Y"	13	20	44	5	147

HF: Hailfinder; PF: Pathfinder

Mshielded "Y": there exists a measured variable that makes the two root nodes of "Y" d-connected.

Global "Y": substructure of a Bayesian network with "Y" as the skeleton irrespective of other arcs.

7.1 OR RESULTS

In this section we present results of application of the OR algorithm to the different datasets.

Table 16: OR precision and recall on different datasets based on global Y arcs (20,000 samples).

Dataset	GYA in output	GYA in both	GYA in generating	GYA_Precision	GYA_Recall
Alarm	17	11	13	0.647	0.846
Hailfinder	16	9	20	0.562	0.450
Barley	21	10	44	0.476	0.227
Pathfinder	35	1	5	0.0286	0.200
Munin	194	41	147	0.211	0.279
Mean				0.3849	0.4004

GYA in output: Total number of GYA present in the Bayesian network output by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

GYA in generating: GYA in the generating network.

Note that precision is particularly low for Pathfinder and Munin (see Table 16).

7.2 PC AND FCI RESULTS

In this section we present results of application of PC and FCI algorithms to the different datasets. FCI could be run only on the Alarm dataset using sample sizes greater than 500. It went out of memory with sample sizes greater than 500 for the Hailfinder dataset. For Barley, Pathfinder and Munin datasets it went out of memory with sample sizes greater than 200. FCI had a precision of 0.57 and a recall of 1.0 on the Alarm dataset.

Table 17: PC precision and recall on different datasets based on global Y arcs (20,000 samples).

Dataset	Total output	GYA in both	GYA in generating	GYA_Precision	GYA_Recall
Alarm	13	12	13	0.923	0.923
Hailfinder	7	6	20	0.857	0.300
Barley	12	12	44	1.00	0.273
Pathfinder	35	0	5	0	0
Munin	26	21	147	0.808	0.143
Mean				0.7176	0.3278

Total output: Total number of arcs output as causal by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

GYA in generating: GYA in the generating network.

Precision and recall for Pathfinder are particularly low (see Table 17).

7.3 BLCD RESULTS

In this section we present results of application of BLCD, BLCDpk, BLCDcv, and the BLCDvss to the different datasets. Section 7.3.1 describes the results based on what is theoretically discoverable for each algorithm which is a type of internal test, and Section 7.3.2 describes the results based on what is theoretically discoverable across PC, FCI, OR, BLCD and BLCD variants, which is more a comparative test. See Section 6.3.2 for a description of what is theoretically discoverable for each algorithm and across all algorithms.

7.3.1 Based on what is theoretically discoverable by the algorithm

BLCD had a recall value of 1.0 for Alarm and Barley. BLCD precision was low for Barley and Pathfinder. Barley has only 2 unshielded Y structures while Pathfinder has none in the generating network. Note that when the prior of the Y structure is low, the positive predictive value (precision) is also likely to be low.

Table 18: BLCD precision and recall based on unshielded and unconfounded Y arcs (20,000 samples).

Dataset	Total output	UYA in both	UYA in generating	UYA_Precision	UYA_Recall
Alarm	15	6	6	0.400	1.00
Hailfinder	8	5	11	0.625	0.455
Barley	7	2	2	0.286	1.00
Pathfinder	2	0	0	0	NA
Munin	43	31	96	0.721	0.323
Mean				0.4064	0.6945

Total output: Total number of arcs output as causal by the algorithm.
UYA: Unshielded and unconfounded Y arc.
UYA in both: UYA output by the algorithm and present in the generating network.
UYA in generating: UYA in the generating network.
NA: Not applicable.

Table 19: BLCDpk precision and recall based on unshielded and unconfounded Y arcs (20,000 samples).

Dataset	Total output	UYA in both	UYA in generating	UYA_Precision	UYA_Recall
Alarm	10	6	6	0.600	1.00
Hailfinder	8	5	11	0.625	0.455
Barley	7	2	2	0.286	1.00
Pathfinder	2	0	0	0	NA
Munin	42	31	96	0.738	0.323
Mean				0.4498	0.6945

Total output: Total number of arcs output as causal by the algorithm.
UYA: Unshielded and unconfounded Y arc.
UYA in both: UYA output by the algorithm and present in the generating network.
UYA in generating: UYA in the generating network.

BLCDpk precision was higher for Alarm but similar to BLCD for the other datasets. Recall values were also similar to BLCD.

BLCDvss precision was the highest for Munin. It also had high recall values for Alarm and Barley. BLCDvss recall for Munin was greater than BLCD and BLCDpk, but its precision was lower.

The performance of BLCDvss and BLCDcv are often poorer than that of BLCD based on both precision and recall on a given dataset. For BLCDvss the recall denominator (SUYA) is higher than that of BLCD (UYA) for most datasets. Even though BLCDvss conditions on Mshielded Y structures and converts them into unshielded Y structures, it fails in situations where the Y structures are shielded by more than one measured variable (see Figures 38 and 39, Y_5 and Y_8).

Table 20: BLCDvss precision and recall based on Mshielded and unshielded but unconfounded Y arcs (20,000 samples).

Dataset	Total output	SUYA in both	SUYA in generating	SUYA_Precision	SUYA_Recall
Alarm	16	6	6	0.375	1.00
Hailfinder	21	5	13	0.238	0.385
Barley	18	2	3	0.111	0.667
Pathfinder	30	0	0	0	NA
Munin	92	48	112	0.522	0.429
Mean				0.2492	0.62025

Total output: Total number of arcs output as causal by the algorithm.

SUYA: Mshielded and unshielded but unconfounded Y arcs.

SUYA in both: SUYA output by the algorithm and present in the generating network.

SUYA in generating: SUYA in the generating network.

Table 21: BLCDcv precision and recall based on unshielded and unconfounded Y arcs (20,000 samples).

Dataset	TO	UYA in both	U_UYA in both	UYA in generating	UYA_P	UYA_R
Alarm	38	15	4	6	0.395	0.667
Hailfinder	39	8	5	11	0.205	0.455
Barley	16	1	2	2	0.0625	1.00
Pathfinder	3	0	0	0	0	NA
Munin	219	61	30	96	0.279	0.312
Mean					0.1883	0.6085

TO: Total number of arcs output as causal by the algorithm.

UYA: Unshielded and unconfounded Y arc.

UYA in both: UYA output by the algorithm and present in the generating network.

U_UYA in both: Unique UYA output by the algorithm and present in the generating network. This was used for UYA_R calculation.

UYA in generating: UYA in the generating network.

UYA_P: UYA Precision; UYA_R: UYA Recall

7.3.2 Based on the union of discoverable causes across all algorithms

7.3.2.1 Precision-recall graphs for each dataset For comparative evaluation of the performance of the different algorithms, GYA precision and GYA recall are plotted. Note that these precision and recall values are based on the union of all Y arcs in the generating structure.

The OR algorithm does not have a parameter that can be used as a threshold to generate precision versus recall graphs. For the Alarm dataset FCI went out of memory with significance levels higher than 0.05 and for the other datasets with sample sizes greater than 500. PC also went out of memory with significance levels higher than 0.05 for Hailfinder, Barley, Pathfinder and Munin. For Alarm, FCI has one entry and for the other datasets none. OR has only one entry for all the datasets while PC has only one entry for Hailfinder, Barley, Pathfinder and Munin based on the default thresholds. Note that these precision-recall graphs can have multiple points for BLCD and BLCD variants because they output probabilities and we have plotted performances at different thresholds. Likewise, for the Alarm precision-recall plot PC also has multiple points as PC could be run with various significance levels on the Alarm dataset.

Figure 40 provides precision versus recall plots at the sample size of 20,000 for the BLCD, BLCDpk, BLCDvss, BLCDcv and PC with the Alarm dataset. Based on what is discoverable across all algorithms, the global algorithms PC and OR, and the local algorithms BLCD and its variants had varying performance ranges in terms of precision and recall. It is notable that BLCD precision is close to that of BLCDpk at the same recall value of 0.77 even though BLCD is not provided with prior knowledge of root nodes.

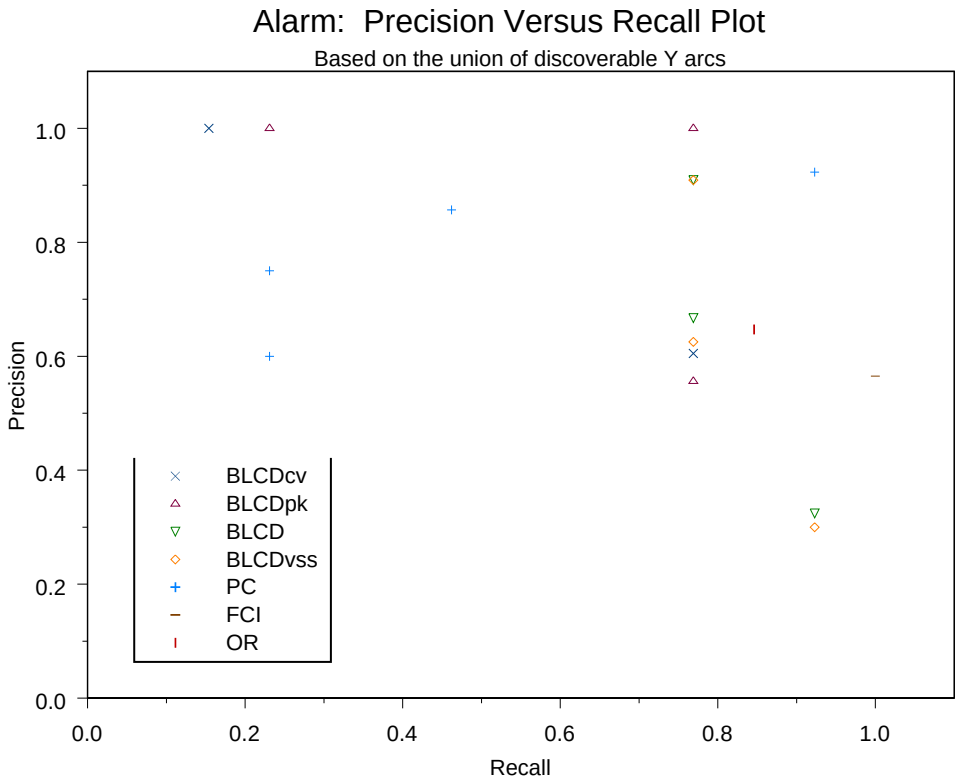


Figure 40: Alarm precision versus recall plot. (20,000 samples)

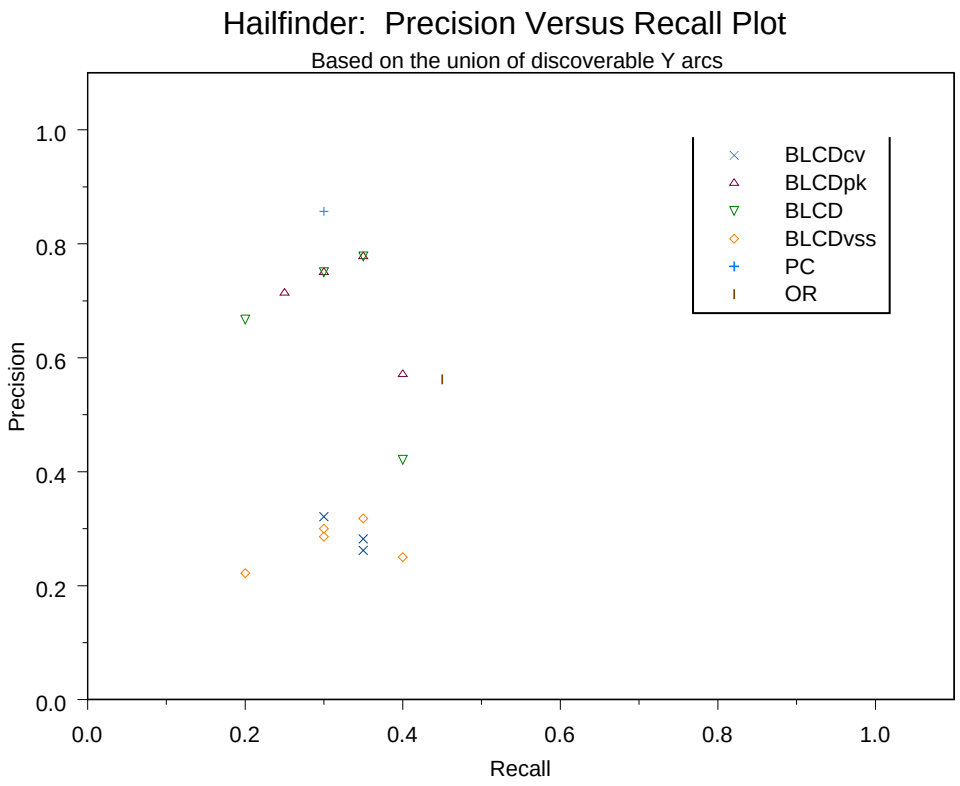


Figure 41: Hailfinder precision versus recall plot. (20,000 samples)

In the Hailfinder dataset (see Figure 41) PC has the best precision (≈ 0.85) with BLCD and BLCDpk close behind with a precision of (≈ 0.79).

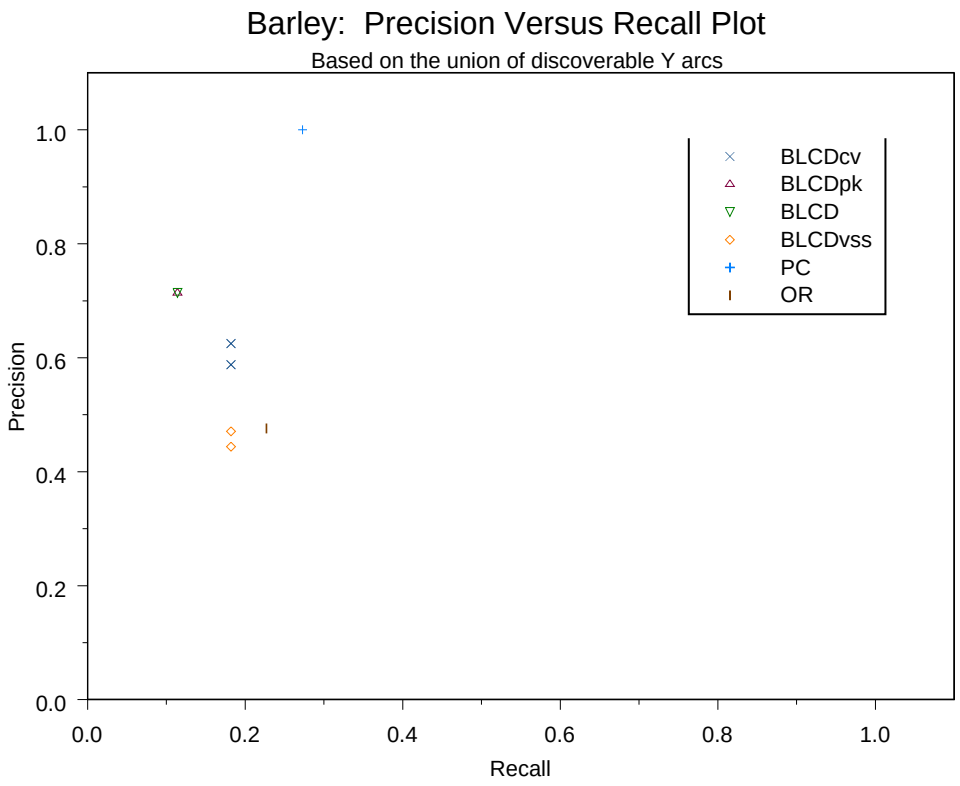


Figure 42: Barley precision versus recall plot. (20,000 samples)

In the Barley dataset (see Figure 42) PC dominates with a precision of 1.0 and a recall of (≈ 0.26).

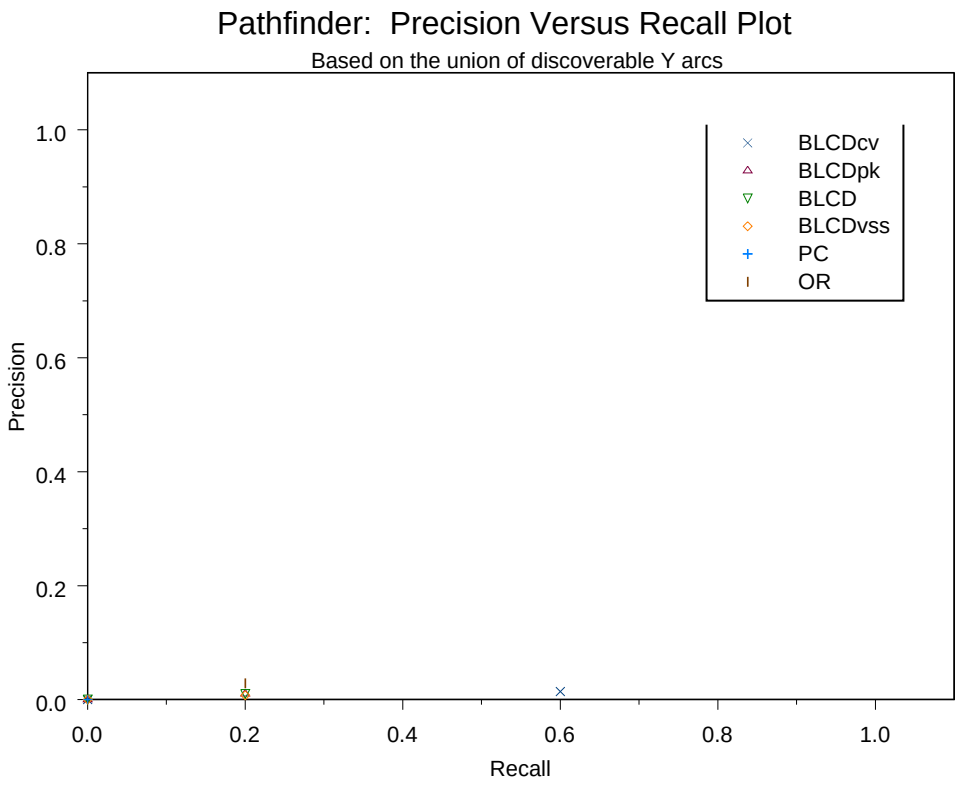


Figure 43: Pathfinder precision versus recall plot. (20,000 samples)

For the Pathfinder dataset (see Figure 43) precision was near zero and recall was low (no Mshielded or unshielded Y structures in the generating network) for BLCD and its variants except BLCDcv. The global algorithms PC and OR did not perform well on this dataset even though there were 5 global Y arcs (GYA).

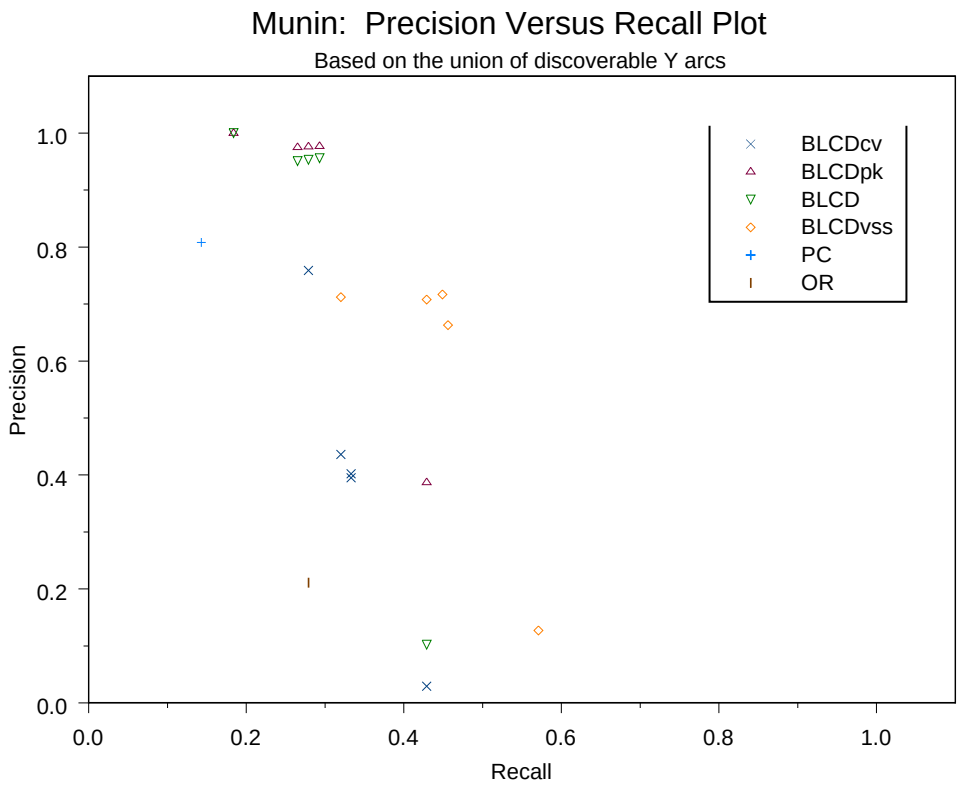


Figure 44: Munin precision versus recall plot. (20,000 samples)

On the Munin dataset also BLCD performance is comparable to BLCDpk (see Figure 44). Both of these algorithms have the best precision (≈ 1.0).

Table 22: BLCD precision and recall based on global Y arcs (20,000 samples).

Dataset	Total output	GYA in both	GYA in generating	GYA_Precision	GYA_Recall
Alarm	15	10	13	0.667	0.769
Hailfinder	8	6	20	0.750	0.300
Barley	7	5	44	0.714	0.114
Pathfinder	2	0	5	0	0
Munin	43	41	147	0.953	0.279
Mean				0.6168	0.2924

Total output: Total number of arcs output as causal by the algorithm.
GYA in both: GYA output by the algorithm and present in the generating network.
GYA in generating: GYA in the generating network.

7.3.2.2 Summary tables for each algorithm For the BLCD class of algorithms there is an increase in precision for all the datasets except Pathfinder when the evaluation is based on the union of discoverable causes. Likewise, there is an overall decrease in recall with this evaluation method.

Table 23: BLCDpk precision and recall based on global Y arcs (20,000 samples).

Dataset	Total output	GYA in both	GYA in generating	GYA_Precision	GYA_Recall
Alarm	10	10	13	1.00	0.769
Hailfinder	8	6	20	0.750	0.300
Barley	7	5	44	0.714	0.114
Pathfinder	2	0	5	0	0
Munin	42	41	147	0.976	0.279
Mean				0.688	0.2924

Total output: Total number of arcs output as causal by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

GYA in generating: GYA in the generating network.

Table 24: BLCDvss precision and recall based on global Y arcs (20,000 samples).

Dataset	Total output	GYA in both	GYA in generating	GYA_Precision	GYA_Recall
Alarm	16	10	13	0.625	0.769
Hailfinder	21	6	20	0.286	0.300
Barley	18	8	44	0.444	0.182
Pathfinder	30	0	5	0	0
Munin	92	66	147	0.717	0.449
Mean				0.4144	0.34

Total output: Total number of arcs output as causal by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

GYA in generating: GYA in the generating network.

Table 25: BLCDcv precision and recall based on global Y arcs (20,000 samples).

Dataset	TO	GYA in both	U_GYA in both	GYA in generating	GYA_P	GYA_R
Alarm	38	23	10	13	0.605	0.769
Hailfinder	39	11	7	20	0.282	0.350
Barley	16	10	8	44	0.625	0.182
Pathfinder	3	0	0	5	0	0
Munin	219	88	49	147	0.402	0.333
Mean					0.3828	0.3268

TO: Total number of arcs output as causal by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

U_GYA in both: Unique GYA output by the algorithm and present in the generating network. This was used for GYA Recall calculation.

GYA in generating: GYA in the generating network.

GYA_P: GYA Precision; GYA_R: GYA Recall

7.3.2.3 A global summary table over all the simulated datasets

In this section we present a summary performance of the direct causal discovery algorithms based on the aggregate number of GYA present in all the data generating networks (see Table 26). Altogether there were 229 GYA.

Table 26: Precision and recall based on global Y arcs from all datasets (20,000 samples).

Algorithm	Total	GYA in both	Unique GYA in both	GYA in generating	GYA_P	GYA_R
OR	283	72	72	229*	0.254	0.314
BLCD	75	62	62	229	0.827	0.271
BLCDcv	315	132	74	229	0.419	0.323
BLCDpk	69	62	62	229	0.899	0.271
BLCDvss	177	90	90	229	0.508	0.393
PC	93	51	51	229	0.548	0.223

*Total number of GYA present in the data generating Alarm, Hailfinder, Barley, Pathfinder and Munin networks.

Total: Total number of arcs output as causal by the algorithm.

GYA in both: GYA output by the algorithm and present in the generating network.

Unique GYA in both: This is different from "GYA in both" for BLCDcv as it combines variables X and Z to output X and/or Z as the cause of Y.

GYA in generating: GYA in the generating network.

GYA_P: GYA Precision; GYA_R: GYA Recall

BLCDpk had the highest precision (0.899) followed by BLCD (0.827). The best recall was achieved by BLCDvss (0.393).

We used a Z test statistic (two sided) to test the difference between the two proportions across all the algorithms pairwise for both precision and recall (algorithm A precision versus algorithm B precision, and algorithm A recall versus algorithm B recall). Standard errors were estimated and 95% confidence intervals (CI) were computed after pooling the two proportions (Daniel, 1991, pages 152 and 225). The null hypothesis of no difference in the two proportions was rejected if the p value was < 0.003 as we did multiple comparisons (15) of precision and recall proportions.

It is observed that there is a significant pairwise difference in the proportions of eleven out of fifteen precision comparisons ($p < 0.0001$). See Table 27 for the details. However, there

Table 27: Dataset aggregation: Precision significance based on all the 229 global Y structures

Algorithm	OR	BLCDpk	BLCD	BLCDvss	PC	BLCDcv
OR		+	+	+	+	+
BLCDpk			−	+	+	+
BLCD				+	+	+
BLCDvss					−	−
PC						−
BLCDcv						

p values of all the significant pairs were < 0.0001.

+: Significant; −: Not significant.

is a significant difference in recall proportions only between BLCDvss and PC ($p < 0.0001$).

7.4 LCD RESULTS

In this section we present results of application of LCDa, LCDb, and LCDc to the datasets.

Table 28: LCDa, LCDb, LCDc precision based on causal and unconfounded pairs (20,000 samples).

Dataset	LcdaT	LcdaC	LcdaCP	LcdbT	LcdbC	LcdbCP	LcdcT	LcdcC	LcdcCP
Alarm	143	50	0.350	122	50	0.410	83	49	0.590
Hailfinder	109	39	0.358	107	39	0.364	103	39	0.379
Barley	125	6	0.0480	86	6	0.0698	84	6	0.0714
Pathfinder	98	1	0.0102	98	1	0.0102	98	1	0.0102
Munin	4791	368	0.0768	2680	362	0.135	1417	337	0.238
Mean			0.1686			0.1978			0.25772

LcdaC, LcdbC, LcdcC: Causal and unconfounded pairs in the output for Lcda, Lcdb and Lcdc respectively, and in the generating network.
LcdaT, LcdbT, LcdcT: Total pairs output as causal and unconfounded by Lcda, Lcdb and Lcdc respectively.
LcdaCP: Lcda Precision; LcdbCP: Lcdb Precision; LcdcCP: Lcdc Precision.

There was only one discoverable causal and unconfounded pair for the Pathfinder network and hence precision was very low for that dataset.

It is clear from the mean precision and recall values that LCDb shows higher precision compared to LCDa. Likewise, LCDc shows higher precision compared to both LCDa and LCDb.

The recall values of LCDa, LCDb and LCDc as shown in Table 29 are comparable. Note that this recall is relative to what LCD is capable of discovering.

Table 29: LCDa, LCDb, LCDc recall based on causal and unconfounded pairs (20,000 samples).

Dataset	CUP	DCUP	aT	aC	aCR	bT	bC	bCR	cT	cC	cCR
Alarm	167	53	143	50	0.943	122	50	0.943	83	49	0.925
Hailfinder	307	109	109	39	0.358	107	39	0.358	103	39	0.358
Barley	206	30	125	6	0.200	86	6	0.200	84	6	0.200
Pathfinder	109	1	98	1	1.00	98	1	1.00	98	1	1.00
Munin	1785	575	4791	368	0.640	2680	362	0.630	1417	337	0.586
Mean					0.6282			0.6262			0.6138

aC, bC, cC: Causal and unconfounded pairs in the output for Lcda, Lcdb and Lcdc respectively, and in the generating network.

CUP: Total number of Causal and unconfounded pairs in the generating network;

DCUP: Total number of Discoverable CUP in the generating network. (See Section 6.3.1 for a description.)

aT, bT, cT: Total pairs output as causal and unconfounded by Lcda, Lcdb and Lcdc respectively.

aCR: Lcda Recall; bCR: Lcdb Recall; cCR: Lcdc Recall.

7.5 INFANT DATASET RESULTS

Table 30 gives the precision and recall for all the algorithms using a sample size of 20,000 based on both the gold and platinum standard categories. Note that for this dataset, precision and recall are based on causal relationships whether confounded or not. For this dataset there is no notion of direct and indirect causal relationships as it is real-world data for which there is no causal Bayesian network described in the literature. Based on the gold standard categorization, OR, PC and LCD have better precision compared to BLCD and its variants. In terms of recall, BLCDvss has the best performance while the LCD algorithms are a close second.

Table 30: Infant: Summary results (20,000 samples).

Algorithm	G.Precision	G.Recall	P.Precision	P.Recall
OR	0.667	0.0468	0.333	0.0238
PC	1.0	0.00585	NA	0
BLCD	0.167	0.0234	0.143	0.0357
BLCDpk	0.133	0.0234	0.125	0.0357
BLCDvss	0.212	0.0819	0.136	0.0714
LCDa	0.565	0.0760	0.533	0.0952
LCDb	0.619	0.0760	0.533	0.0952
LCDc	0.619	0.0760	0.533	0.0952

G: Gold standard; P: Platinum standard.

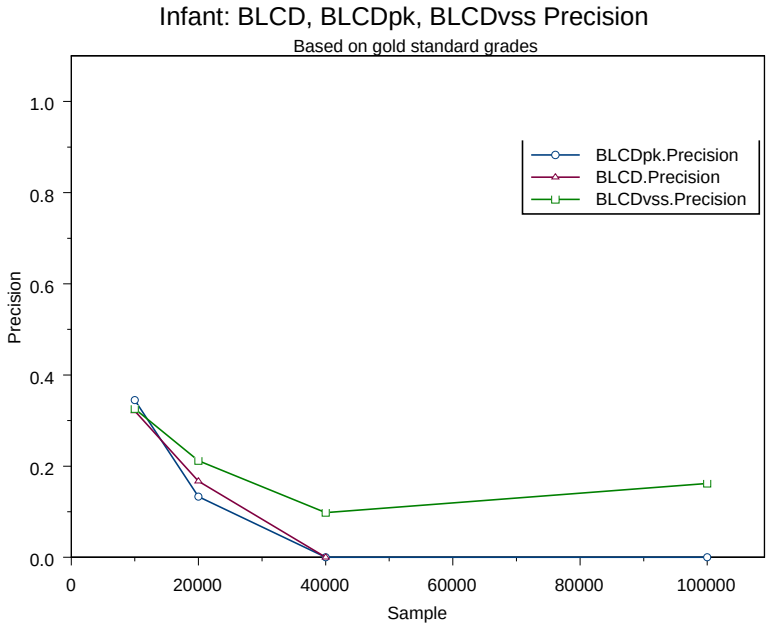
In Figures 45– 47, we plot precision and recall trends for the various algorithms based on the gold standard categorization as sample size increases from 10,000 to 100,000. Figure 45 presents a comparison of the results of BLCD and LCD based on precision as sample size increases from 10,000 instances to 100,000. BLCD precision initially reaches 0.32, then unexpectedly and paradoxically drops to 0.17 at a sample size of 20,000 and becomes 0 at the sample size of 40,000. No output is produced at the sample size of 100,000. The precision values for BLCDpk are 0.34 at the sample size of 10,000 and 0.13 at the sample size of 20,000. BLCDpk precision is 0 at higher sample sizes. BLCDvss has a precision of 0.33 at the sample size of 10,000, then falls to 0.21 at the sample size of 20,000, drops

to 0.1 at 40,000 and then climbs to 0.16 at the sample size of 100,000. It is interesting that by allowing conditioning the precision does not go to zero. This suggests that perhaps Mshielded Y structures are being detected at larger sample sizes. The performance of LCD is better with LCDa precision falling in the range of 0.60 and 0.41. LCDb and LCDc show similar precision values in the range of 0.60 and 0.24.

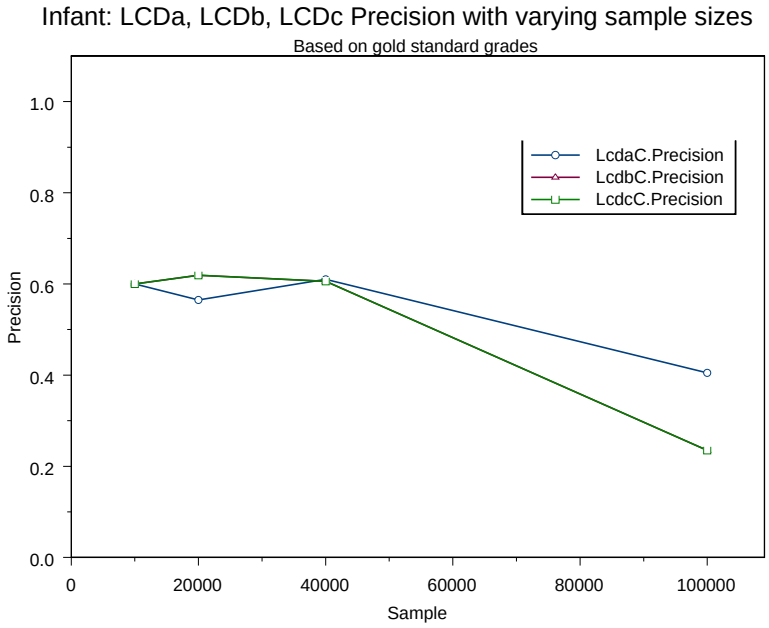
Figure 46 presents a comparison of the results of BLCD and LCD based on recall as sample size increases from 10,000 instances to 100,000. BLCD recall is 0.12 at the sample size of 10,000 and paradoxically goes to 0 at the sample size of 40,000. The recall values of BLCDpk are similar. BLCDvss recall values fall in the range of 0.16 and 0.07. The recall values of LCDa, LCDb and LCDc are higher and reach a maximum at the sample size of 40,000, but then paradoxically the recall drops at a sample size of 100,000.

Figure 47 presents a comparison of the results of OR and PC based on precision and recall as sample size increases from 10,000 instances to 100,000 using the gold standard.

The precision of OR has a peak at the sample size of 20,000 while OR recall improves with increasing sample sizes. The precision of PC is 1.0 at the sample size of 20,000 and it is 0 at other sample sizes. PC recall is close to zero. PC went out of memory with a sample size of 100,000.

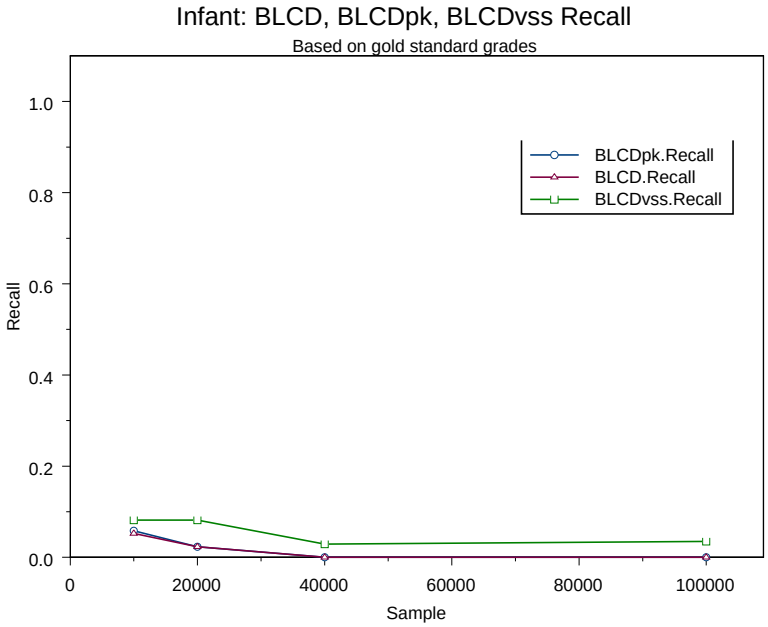


45.1: BLCD Precision

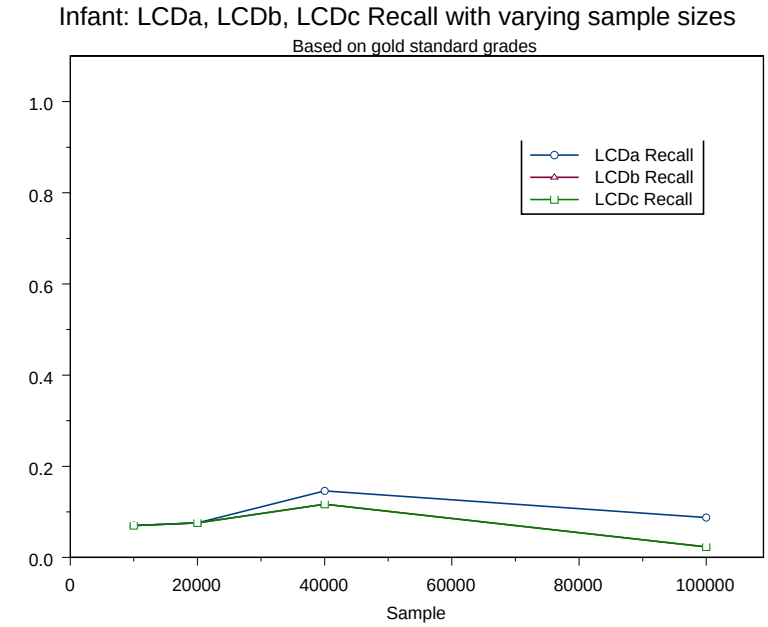


45.2: LCD Precision

Figure 45: BLCD and LCD precision result on Infant data: Figure 45.1 describes BLCD precision; Figure 45.2 describes LCD precision.

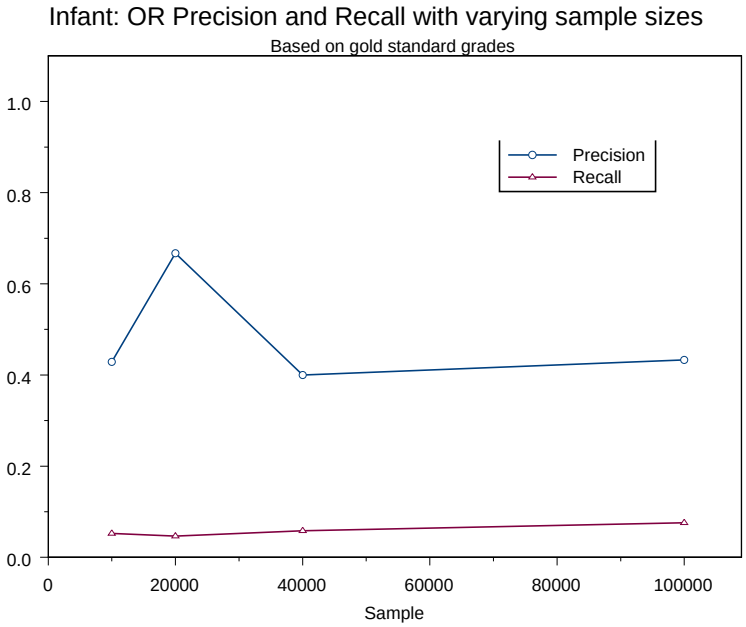


46.1: BLCD Recall

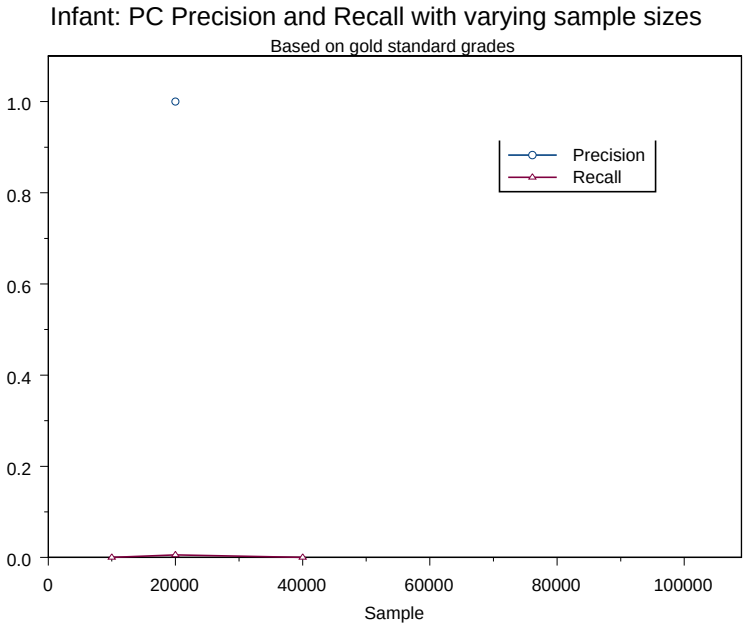


46.2: LCD Recall

Figure 46: BLCD and LCD recall result on Infant data: Figure 46.1 describes BLCD recall; Figure 46.2 describes LCD recall.



47.1: OR



47.2: PC

Figure 47: OR and PC result on Infant data: Figure 47.1 describes OR result; Figure 47.2 describes PC result.

7.6 RUNTIMES

Table 31: Algorithm runtimes in seconds for the different datasets.

Algorithm	Alarm	Hailfinder	Barley	Pathfinder	Munin	Infant
OR	37768	105292	7338	97072	21023	67089
FCI	53	–	–	–	–	–
PC	21	881	328	10978	1604	7803
BLCD	248	1045	16764	3865	41919	6009
BLCDpkvsscv	1409	7454	29804	42502	169644	132456
LCDabc	261	121	218	6058	21921	1957

BLCDpkvsscv: BLCDpk, BLCDvss and BLCDcv were run together.

– FCI went out of memory.

PC could be run only on 40,000 samples of Infant dataset.

Infant sample size 100,000; other datasets 20,000.

Table 31 gives the computation times of the various algorithms. PC had the least runtime for Alarm and Munin, LCD for Hailfinder, Barley and Infant, and BLCD for Pathfinder. FCI could be run only on the Alarm dataset and went out of memory with all the other datasets.

8.0 DISCUSSION

In this Chapter we discuss the results of the various experimental runs described in Chapter 7 and present the implications of our research for discovering causal relationships from observational data. The interpretation of the results is based on the hypothesis that was presented in detail in Section 1.2 and summarized below for reference.

Hypothesis: *Causal discovery using local search methods in context $\mathcal{C}$ will have better performance compared to the causal discovery methods using global search as described in Section 3.*

The context $\mathcal{C}$ for the causal data mining hypothesis incorporates *large datasets, an anytime framework and limited computational resources.* Performance evaluation was done based on the measures of precision and recall.

All the algorithms could be run to completion on the Alarm data set. The FCI could not be run on the other datasets as the implementation that was available to me did not scale up and went out of memory. Also PC did not scale up on the Infant dataset sample size of 100,000. The OR, BLCD and LCD algorithms ran to completion on all the datasets and sample sizes tested.

Note that OR, BLCD, BLCDpk, BLCDvss, BLCDcv, PC and FCI output "direct causal relationships" or arcs of the generating causal structure. More precisely, the BLCD algorithms output unconfounded direct causal relationships. For comparison across these algorithms precision and recall values were computed based on the the union of what is discoverable from the generating structure.

The LCD algorithms (LCDa, LCDb, LCDc) output unconfounded direct and indirect causal relationships. Note that a direct comparison of LCD algorithms with the algorithms that output direct causal relationships is not possible.

Table 32: Alarm to Munin: Based on what is theoretically discoverable for each algorithm

Algorithm	Mean.Precision	Std.Error	CI	Mean.Recall	Std.Error	CI
OR	0.385	0.115	0.065–0.705	0.400	0.120	0.068–0.732
BLCD	0.406	0.128	0.052–0.761	0.695	0.178	0.127–1.262
BLCDcv*	0.188	0.072	-0.010–0.387	0.609	0.149	0.133–1.084
PC*	0.718	0.182	0.212–1.224	0.328	0.158	-0.111–0.767
BLCDpk	0.450	0.135	0.074–0.825	0.695	0.178	0.127–1.262
BLCDvss	0.249	0.093	-0.008–0.506	0.620	0.141	0.172–1.069

CI: Confidence interval

Overall differences in precision means were significant ($p = 0.002$).

*Pairwise comparisons of precision means were statistically significant only between BLCDcv and PC ($p = 0.001$).

Overall differences in recall means were not statistically significant ($p = 0.08$).

Table 33: Alarm to Munin: Based on the union of what is discoverable over all the algorithms (global Y structures)

Algorithm	Mean.Precision	Std.Error	CI	Mean.Recall	Std.Error	CI
OR	0.385	0.115	0.065–0.705	0.400	0.120	0.068–0.732
BLCD	0.617	0.162	0.168–1.066	0.292	0.131	-0.072–0.657
BLCDcv	0.383	0.115	0.063–0.702	0.327	0.127	-0.027–0.680
PC	0.718	0.182	0.212–1.224	0.328	0.158	-0.111–0.767
BLCDpk	0.688	0.181	0.184–1.192	0.292	0.131	-0.072–0.657
BLCDvss	0.414	0.127	0.060–0.768	0.340	0.130	-0.021–0.701

CI: Confidence interval

Overall differences in precision means were statistically significant ($p = 0.005$).

Pairwise comparisons of precision means were not statistically significant.

Differences in recall means were not significant ($p = 0.57$).

Tables 32 and 33 provide mean precision and recall with standard errors and confidence intervals over all the simulated datasets for all the algorithms that output direct causal relationships. To assess the significant difference in both precision and recall across the different algorithms as a whole after adjusting for datasets, 2 way ANOVA was carried out. Bonferroni multiple comparison method (Armitage et al., 2002; Miller, 1981; Kleinbaum, 1998) was used to look for specific differences in adjusted means of precision and recall between pairs of algorithms if there was a significant difference in the means as a whole. All multiple comparison tests used a 2-tailed significance level of $p < 0.003(0.05/15)$[1]. All tests were performed using SAS software, version 8.0 (SAS Institute, Cary, NC).

Based on what is theoretically discoverable by each algorithm (See Table 32), for the overall test statistic, it was found that there was a statistically significant difference in the adjusted precision means for at least one algorithm to yield higher values than at least one of the other algorithms ($p < 0.002$). There was a significant pairwise difference observed in the precision means between BLCDcv and PC ($p = 0.001$). We did not find any significant difference in the adjusted recall means across the 6 original algorithms (p=0.08).

Based on the union of what is discoverable across all the different algorithms that output direct causal relationships (See Table 33), there was a significant difference in the adjusted precision means across the algorithms ($p = 0.005$), but no significant pairwise differences were noted at $p < 0.003$. There was no significant difference in the adjusted recall means across all the algorithms ($p = 0.57$).

The statistical power of the tests were low as they were based on mean precision and recall values for the different algorithms obtained from five datasets.

[1]We divide 0.05 by 15 because 15 pairwise comparisons were done for the six algorithms.

8.1 ALARM

Only BLCD, BLCDpk, and BLCDvss algorithms are able to attain full recall (UYA/SUYA recall of 1.0) on this dataset based on what is theoretically discoverable. Only FCI attained a GYA recall of 1.0. PC had the best precision among all the algorithms that were not given any additional knowledge as input.

8.2 HAILFINDER

When compared to the Alarm dataset, both the global and local algorithms had low recall with the Hailfinder dataset.

8.3 BARLEY

The degradation in performance of LCDa, LCDb and LCDc algorithms on this dataset compared with Alarm and Hailfinder can be explained by the lower proportion of unconfounded causal pairs (see Table 7). The proportion of the discoverable causal and unconfounded pairs (DCUP) is also considerably lower (see Table 28). Note that when the prior of the discoverable causal and unconfounded pairs is low, the positive predictive value is also likely to be low.

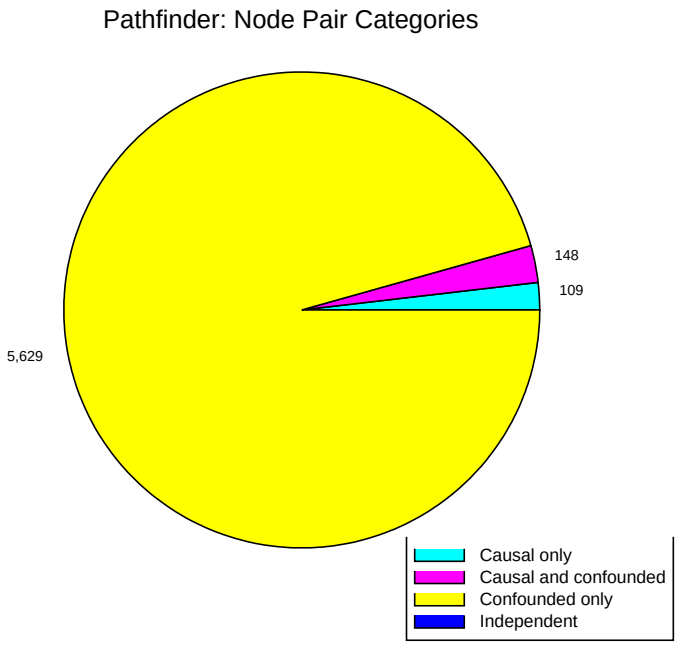
8.4 PATHFINDER

Pathfinder also has lower proportion of unconfounded causal arcs (see Table 8) compared to Alarm and Hailfinder. Pathfinder does not have any "Y" structures (see Table 15), either Mshielded or unshielded and has just one root node. The number of GYA is also low (5). Moreover, 96% of the pairwise relationships in this network belong to the confounded-only

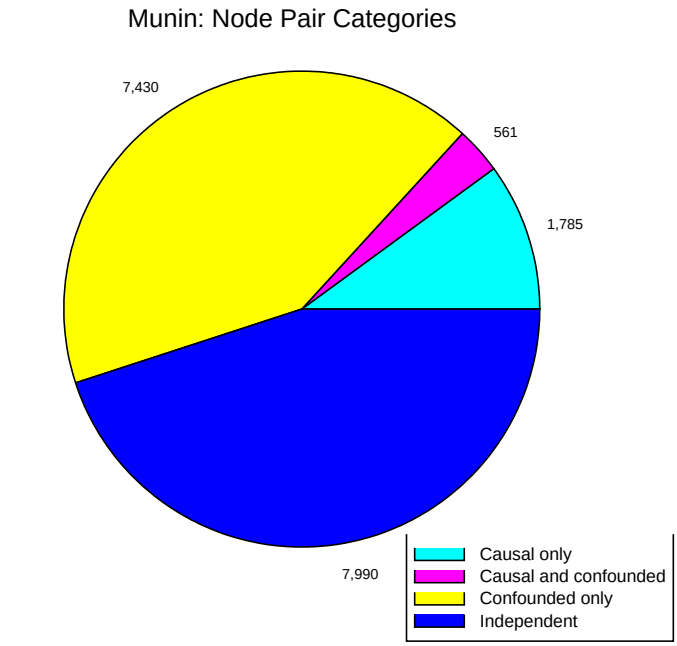
(COP) category and there are no independent pairs. Figure 48 provides a comparison of the node pair categories of Pathfinder and Munin networks. All these factors may have contributed to the poor performance of the various algorithms on this dataset.

8.5 MUNIN

Munin has a high proportion of unconfounded causal arcs (see Table 8) similar to Alarm and Hailfinder. Munin also has a large number of "Y" structures (see Table 15). The Munin network also has 34 root nodes that contribute to the better performance of the various algorithms, in particular the LCD algorithms and BLCDpk that take as input the root nodes. Note that root nodes make "Y" structures more likely and this property helps BLCD, BLCDpk and BLCDvss. BLCDvss had the best recall identifying 66 out of the 147 GYA in the data generating network.



48.1: Pathfinder node pair categories



48.2: Munin node pair categories

Figure 48: Pair categorization for Pathfinder and Munin: Figure 48.1 refers to the Pathfinder network; Figure 48.2 refers to the Munin network.

8.6 INFANT

The following discussion is based on the evaluation using the Gold standard. Platinum standard performance measures were comparable. BLCD and BLCDpk precision are at their highest point at the sample size of 10,000 and gradually reduce to 0 at the sample size of 40,000. The precision for BLCDvss starts lower than BLCDpk at the sample size of 10,000 but stays higher when compared to BLCD and BLCDpk. The precision for LCD algorithms is ≈ 0.6 in the sample range of 10,000 to 40,000 and then gradually start falling with increasing sample size.

The recall for BLCD and BLCDpk also shows a similar degradation in performance starting higher at the sample size of 10,000 and touching 0 at the sample size of 40,000. The recall of BLCDvss is maximal at the sample size of 20,000 and then it gradually goes down to 0.07. LCD recall is maximal at the sample size of 40,000 and then gradually comes down.

There is a general degradation in performance with increasing sample sizes for both BLCD and LCD algorithms once the 20,000 to 40,000 range is reached. This might be because detection of hidden confounding occurs more at larger sample sizes. This could result in Y structures becoming confounded, Mshielded or both. This increased detection of Mshielded Y structures (instead of unshielded Y structures) explains the relatively better performance of the BLCDvss. However, further exploration is required to understand this phenomenon clearly.

The performance of OR and LCD are comparatively better in terms of both precision and recall.

As stated in Chapter 7 the performance on Infant data of BLCDvss and of OR seem to suggest that Bayesian conditioning on measured confounders is helpful in causal discovery.

8.7 IMPLICATIONS FOR CAUSAL DISCOVERY

This dissertation research has highlighted the role of Y structures for causal discovery from observational data using global and local causal Bayesian network learning algorithms. The

study addressed different types of Y structures—unshielded, Mshielded, global, and their role in causal discovery using the different algorithms.

Even though there is no clear winner in terms of performance, PC and the BLCD class of algorithms seem to have better precision compared to OR with simulated datasets. This is based on the default thresholds for the algorithms tested. This implies less number of false positives for PC, BLCD, BLCDvk and BLCDvss when compared to OR. A desirable goal of causal datamining is to keep the proportion of false positives low even if it entails a trade-off in terms of recall.

Though BLCDpk has additional background knowledge than BLCD, it only performs better in a few instances (Alarm and Munin GYA_Precision). This result suggests that it may be possible to learn causal relationships without assuming root nodes, almost as well as having them.

When performance evaluation was based on the pooled GYA from all the data generating networks, BLCD and BLCDpk had much higher precision compared to the global algorithms, while BLCDvss had the best recall (see Table 26).

When the order of variables is ≈ 100, the global algorithms are able to handle sample sizes of 20,000 instances well. If the sample sizes are higher, the global algorithms can work with subsamples for causal discovery. However, if the number of variables are considerably more than the datasets that were used in this study (for example, gene expression datasets that have ten thousand variables) the global algorithms will almost certainly have to use subsets of variables, essentially employing a local causal discovery framework.

The computation times of the various algorithms are given in Table 31. As the complexity of the networks increase from Alarm to Munin, the computation times also increase in general. BLCDpk, BLCDvss and BLCDcv algorithms were run bundled together and hence the runtimes of these algorithms are generally an order of magnitude greater than BLCD. The BLCDcv component is responsible for this increase in runtime.

The global discovery algorithms OR, PC and FCI used all the observed variables for model building and model selection. This confers a considerable advantage to the global methods. On the other hand, by design the local discovery algorithms LCD and BLCD (and their variants) have access to only a small subset of variables at a time (3 for LCD and 4 for

BLCD) in the model evaluation stage. LCD evaluates triplets of variables W, X, Y. BLCD evaluates sets of four variables. Using these four variables, BLCD ascertains the causal influence of a variable X on variable Y.

The performance on Infant data of BLCDvss and of OR seem to suggest that Bayesian conditioning on measured confounders is helpful in causal discovery in this real dataset.

The post-processing of PC output to identify the GYA conferred good precision and recall with most of the network datasets. Note that PC could be run on all network datasets and infant data with a sample size of 40,000. On the other hand, FCI could be run only on the Alarm dataset with a sample size greater than 500. This makes PC with post-processing for Y structures an attractive causal discovery algorithm, particularly in comparison to the current implementation of FCI.

Evaluation of the performance of the global and local causal discovery algorithms that output direct causal relationships based on the union of what is discoverable over all the algorithms (GYA) did not reveal any pairwise differences between the algorithms (with the exception of PC and BLCDcv for precision) in terms of precision and recall for all the network datasets (see Tables 32 and 33). This suggests that with larger datasets (a very large number of variables, a very large number of instances, or both) the local causal discovery algorithms can be expected to have performances in terms of precision and recall comparable to what we would get if the global algorithms could be run to completion on the larger datasets.

Table 34: Effect of combining PC and BLCD on Munin dataset (20,000 instances)

Algorithm	Total	GYA in both	GYA in generating	GYA_P	GYA_R
PC	26	21	147	0.808	0.143
BLCD	43	41	147	0.953	0.279
PC-BLCD	58	51	147	0.879	0.347

Total: Total number of arcs output as causal by the algorithm.
GYA in both: GYA output by the algorithm and present in the generating network.
GYA in generating: GYA in the generating network.
GYA_P: GYA Precision; GYA_R: GYA Recall

From a practical approach to causal discovery our performance results for OR, PC, BLCD

and its variants provide a framework for integrating both global and local algorithms. We could use one global and one local (for example, PC with post-processing and BLCD) and obtain the union of the causal relationships output. Table 34 shows the effect of combining the output of PC and BLCD for the Munin dataset. The PC-BLCD union resulted in a higher recall with a marginal decrease in precision. However, if the goal is to increase precision, two algorithms can be combined by taking the intersection of the output. Naturally this is likely to result in a lower recall.

Though BLCD and its variants are designed to output only unconfounded Y arcs, they also seem to do reasonably well when the evaluation is based on the union of discoverable causes across all algorithms (GYA). This suggests that when BLCD and its variants output Y arcs that have measured confounders, the confounding is likely to be minimal. As BLCD and BLCD variants get credit for discovering such "minimally confounded" Y structures when the evaluation is based on GYA, precision is higher.

The causal discovery framework that we presented for identifying direct causal relationships is dependent on the presence of Y structures in the data generating process. The three medical (Alarm, Pathfinder, Munin) and two non-medical (Hailfinder, Barley) networks that were used to generate data had varying numbers of Y structures. Since these networks were created by domain experts capturing the probabilistic dependencies and independencies in the domain, it is plausible to assume the occurrence of Y structures as components of the data generating process in many real-world domains.

The local algorithms did not dominate the global algorithms in terms of precision, recall and runtime. However, the LCD algorithms did perform relatively well in terms of precision and recall on the one real dataset evaluated, and this deserves follow up with additional real datasets. However, the LCD algorithms require extra knowledge as input in the form of root nodes that serve as a type of instrumental variables.

8.8 CONTRIBUTIONS

In this section we enumerate the salient contributions of this dissertation research.

1. We formalize the task of causal discovery from observational data using a Bayesian approach and local search, and identify *sufficient* structures for causal discovery.
2. We identify the Y structure as sufficient for assigning causality from observational data. We also provide a formal proof for it.
3. We enumerate, categorize and describe the various types of Y structures that can be encountered in the data generating networks. Specifically, we identify all types of Y structures present in the Alarm, Hailfinder, Barley, Pathfinder and Munin networks and classify them.
4. We describe the types of causal structures that are appropriate to seek (for example, GYA when using PC, FCI, OR, and YA with BLCD and BLCD variants) when global and local search methods are employed for causal discovery from observational data.
5. We developed the Bayesian local causal discovery framework and presented a set of algorithms that use this framework.
6. We developed a post-processing method for global Bayesian network learning algorithms thereby extending them to be causal discovery algorithms.
7. We showed that the BLCD class of algorithms that use local search methods perform as well as the global search algorithms in different datasets, implying that the local search algorithms will have comparably good performance on very large datasets when the global methods fail to scale up.
8. We showed principled methods of combining global and local causal discovery algorithms to improve upon the performance of the individual algorithms.

8.9 LIMITATIONS

There are two main types of limitations for this work. The first set of limitations result from the framework and assumptions we have chosen for causal discovery. The second set of

limitations is due to the specifics of the algorithms and the experimental methods that were used. We describe these limitations below.

8.9.1 Causal discovery framework limitations

We have used the framework of causal Bayesian networks (CBNs) to model causality. The CBN framework imposes a directed acyclic graph structure to all causal phenomena. Naturally this excludes causal mechanisms that explicitly incorporate feedback cycles. The Markov and faithfulness assumptions made in the CBN framework require that the data represent the probabilistic dependence and independence relationships implied by the network structure.

8.9.2 Specific algorithmic and experimental methodological limitations

The causal discovery approach we have taken (BLCD approach) is not complete in the sense that it can discover only causal relationships represented in nature as an unconfounded Y structure. The BLCD framework does not model hidden variables explicitly but incorporates them implicitly when models over measured variables are scored. BLCD also currently requires that the modeled variables are discrete.

For the real-world infant mortality dataset we used there was no data-generating structure available to serve as a reference gold standard for scoring the purported causal relationships output by the different algorithms. Two domain experts scored the output of the algorithms and provided the reference standard. However, there is uncertainty in their grading as all the causal relationships in the domain are not clearly known and there is likely to be subjectivity in their grading based on their perception of the relationships among the domain variables. The experts were not asked to differentiate the causal relationships into direct, indirect, causal and unconfounded or causal and confounded. Also, we used only one real-world dataset to validate the different algorithms.

The evaluation measures of precision and recall that were used are structural. Hence the evaluation of the purported causal relationships were structural, leaving out the parametric components.

BLCD and BLCD variants were compared with only three global algorithms OR, PC and FCI.

We define an extended Y structure before introducing another limitation. Assume that the set of arcs $A \rightarrow C, B \rightarrow C, C \rightarrow D$ form a Y structure A, B, C, D. If there is an arc $D \rightarrow E$, we call the resulting structure ABCDE an "extended Y structure". With reference to Figure 36, $B1, B2, B3, B4, B5$ form an extended Y structure. We refer to the $D \rightarrow E$ and the $B4 \rightarrow B5$ arcs as extended Y arcs (EYA). In our evaluation we do not look for extended Y structures. Hence, even though OR, PC and FCI can output the $B4 \rightarrow B5$ arc, they are not given credit for the discovery. The current BLCD framework is not capable of discovering EYA[2].

We plan to address these limitations as part of our future research.

8.10 FUTURE WORK AND OPEN PROBLEMS

Even though the BLCD algorithms in general have high precision values, the recall is generally low. It would be useful to develop algorithmic extensions to the BLCD class of algorithms to improve recall while retaining good precision values. Currently BLCDcv does not distinguish between cause and covariate when it outputs a relationship such as $X \times Z$ causally influences Y. In such a situation we do not know if X is causal, Z is causal or both are causal. It seems worthwhile to explore methods to distinguish between these hypotheses. It also seems useful to combine BLCDpk and BLCDvss, as well as BLCDvss and BLCDcv.

The post-processing of the OR-generated network (to identify Y structures) yielded purported causal arcs. But these causal arcs were based on just one network structure that was output as the "best" based on a Bayesian score. Instead of using just one selected global network for identifying the Y structures, we could do model averaging of the Y structures using many different network models that are weighted by the (estimated) posterior probability of each model. Since OR performed well in the experiments here (albeit often with

[2]We note that by making the node $B4$ hidden, BLCD will discover the $B3 \rightarrow B5$ arc, and using this result the $B4 \rightarrow B5$ could be inferred.

longer run times), it seems that such an extension is worthwhile.

More generally, this research has opened up the potential for a probabilistic version of any global score-based Bayesian network learning algorithm for causal discovery. By identifying the Y structures in different global models and weighting them by their scores, we can convert these algorithms to effective causal discovery algorithms. Even though the resulting algorithms are not necessarily local, they could be made local by using a subset of variables for Bayesian network learning. This direction of research seems quite promising.

In real-world datasets, many variables remain unmeasured or "hidden". Hence it will be useful to evaluate the performance of the global discovery methods by randomly withholding a progressively larger subset of variables as hidden from the algorithms during evaluation. It also would be interesting to apply the causal discovery algorithms to other real-world datasets, particularly medical datasets and compare the performance with the Infant dataset. An important open problem is to construct in a systematic way a larger, integrated causal model using the discovered pairwise causal influences.

In this research we have shown that identification of unconfounded Y structures in a local discovery framework is a sufficient condition for discovering causal relationships from observational data (see Appendix C for related proofs). Likewise, post-processing the output of a global Bayesian network learner to identify the GYA structures (unconfounded and Mconfounded Y structures) is a sufficient condition for causal discovery. In other words we have shown that Y structure identification is a sufficient condition for causal discovery without the assumption of causal sufficiency.

We conjecture that using a CBN framework under Markov and faithfulness assumptions, a Y structure is a necessary condition for causal discovery from observational data without assuming any prior knowledge. We would like to explore formal proofs of this statement as part of our future work.

APPENDIX A

ADDITIONAL RESULTS

In this Chapter we present additional results of the experimental runs described in Chapter 6. All the algorithms except FCI were run with incremental sample sizes ranging from 100 to 20,000 on all the datasets. FCI could be run only on the Alarm dataset and went out of memory with sample sizes above 200 on the other datasets. Hence FCI results are reported only for the Alarm dataset.

Table 35: Alarm: Precision based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0.750	1.00	0.667	0.625	NA	NA	0.400
200	0.667	0.727	0.667	0.400	NA	0	0.400
500	0.579	0.667	0.667	0.440	0.500	0.500	0.889
1000	0.533	0.846	0.667	0.500	0.833	0.583	0.714
2000	0.667	1.00	0.667	0.579	1.00	0.562	0.800
5000	0.708	1.00	0.667	0.688	0.818	0.542	0.857
10000	0.636	1.00	0.667	0.688	1.00	0.522	0.800
20000	0.605	1.00	0.667	0.625	0.923	0.565	0.647

NA: not available.

It is interesting to note that for the Alarm dataset only FCI is able to attain a recall of 1.0. BLCDpk is able to attain a precision of 1.0 but not complete recall.

Barley has ten root nodes. Provding that as prior knowledge did not improve the performance of BLCDpk when compared with BLCD.

Table 36: Alarm: Recall based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0.154	0.462	0.769	0.769	0	0	0.615
200	0.308	0.615	0.769	0.769	0	0	0.615
500	0.462	0.769	0.769	0.846	0.0769	0.231	0.615
1000	0.615	0.846	0.769	0.846	0.385	0.538	0.769
2000	0.615	0.769	0.769	0.846	0.538	0.692	0.923
5000	0.615	0.769	0.769	0.846	0.692	1.00	0.923
10000	0.692	0.769	0.769	0.846	0.923	0.923	0.923
20000	0.769	0.769	0.769	0.769	0.923	1.00	0.846

Table 37: Hailfinder: Precision based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	NA	NA	NA	NA	1.00	1.00	0
200	NA	1.00	1.00	0.200	1.00	1.00	0.500
500	NA	1.00	1.00	0.267	0.750	0.600	0
1000	1.00	0.500	0.500	0.0588	0.750	NA	0.500
2000	NA	0.500	0.667	0.143	0.625	NA	0.875
5000	0.333	0.500	0.500	0.143	0.750	NA	0.625
10000	0.333	0.714	0.714	0.263	0.667	NA	0.692
20000	0.282	0.750	0.750	0.286	0.857	NA	0.562

NA: not available.

Table 38: Hailfinder: Recall based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0	0	0	0	0.150	0.100	0
200	0	0.0500	0.0500	0.0500	0.100	0.100	0.0500
500	0	0.0500	0.0500	0.200	0.150	0.150	0
1000	0.0500	0.150	0.0500	0.0500	0.150	NA	0.0500
2000	0	0.150	0.100	0.100	0.250	NA	0.350
5000	0.100	0.150	0.150	0.150	0.300	NA	0.500
10000	0.250	0.250	0.250	0.250	0.200	NA	0.450
20000	0.350	0.300	0.300	0.300	0.300	NA	0.450

NA: not available.

Table 39: Barley: Precision based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	NA	NA	NA	NA	NA	NA	0.333
200	NA	0.667	0.667	0.667	NA	NA	0
500	0	0.333	0.333	0.400	NA	NA	0.222
1000	NA	0.667	0.667	0.667	1.00	NA	0.667
2000	NA	1.00	1.00	0.625	0.667	NA	0.333
5000	0.333	0.800	0.800	0.533	1.00	NA	0.444
10000	0.500	1.00	1.00	0.625	1.00	NA	0.667
20000	0.625	0.714	0.714	0.444	1.00	NA	0.476

NA: not available.

Table 40: Barley: Recall based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0	0	0	0	0	0	0.0227
200	0	0.0455	0.0455	0.0455	0	0	0
500	0	0.0227	0.0227	0.0455	0	NA	0.0455
1000	0	0.0455	0.0455	0.0909	0.0227	NA	0.0909
2000	0	0.0682	0.0682	0.114	0.0455	NA	0.114
5000	0.136	0.0909	0.0909	0.182	0.114	NA	0.0909
10000	0.136	0.0682	0.0682	0.114	0.114	NA	0.136
20000	0.182	0.114	0.114	0.182	0.273	NA	0.227

NA: not available.

Table 41: Pathfinder: Precision based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0	0	0	0	0	NA	0
200	0	0	0	0	NA	0	0
500	0	0	0	0	0	NA	0.0130
1000	0	0	1.00	0.0244	0	NA	0.0256
2000	0	0	0	0.0135	0	NA	0.0278
5000	0	0	0	0.0143	0	NA	0
10000	0	0	0	0	0	NA	0.0312
20000	0	0	0	0	0	NA	0.0286

NA: not available.

Table 42: Pathfinder: Recall based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0.200
1000	0	0	0.200	0.400	0	NA	0.400
2000	0	0	0	0.200	0	NA	0.400
5000	0	0	0	0.200	0	NA	0
10000	0	0	0	0	0	NA	0.200
20000	0	0	0	0	0	NA	0.200

NA: not available.

For the Pathfinder dataset, the performance of the global algorithms (PC and OR) was better than BLCD and its variants.

On the Munin dataset BLCDpk, BLCD and PC perform well based on precision and BLCDvss has the best recall.

Table 43: Munin: Precision based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	1.00	0.643	0.600	0.560	NA	NA	0.117
200	0.375	0.652	0.706	0.571	NA	NA	0.154
500	0.667	0.833	0.862	0.639	1.00	0.778	0.160
1000	0.543	0.806	0.806	0.624	0.857	NA	0.229
2000	0.736	0.882	0.882	0.618	1.00	NA	0.144
5000	0.513	0.900	0.897	0.649	0.941	NA	1.00
10000	0.514	0.949	0.949	0.677	0.952	NA	0.268
20000	0.402	0.976	0.953	0.717	0.808	NA	0.211

NA: not available.

Table 44: Munin: Recall based on global Y arcs with increasing sample sizes.

Sample	BLCDcv	BLCDpk	BLCD	BLCDvss	PC	FCI	OR
100	0.0136	0.0612	0.0612	0.0952	0	0	0.177
200	0.0204	0.102	0.0816	0.136	0	0	0.129
500	0.109	0.170	0.170	0.313	0.0136	0.0476	0.163
1000	0.156	0.197	0.197	0.361	0.0408	NA	0.184
2000	0.204	0.204	0.204	0.374	0.0544	NA	0.231
5000	0.279	0.245	0.238	0.429	0.109	NA	0.0136
10000	0.299	0.252	0.252	0.456	0.136	NA	0.259
20000	0.333	0.279	0.279	0.449	0.143	NA	0.279

NA: not available.

APPENDIX B

BLCD EQUATION

The equations in this section are provided as a reference for the Y structure proofs.

Under the assumptions made in Chapter 5, the following equation provides a lower bound on the probability of an unconfounded causal relationship between X and Y:

$$\mathrm{P}(X \rightarrow Y|\mathrm{D}) \quad \geq \quad \frac{\mathrm{Score}(\mathrm{G}_1|\mathrm{D})}{\sum_{i=1}^{543} \mathrm{Score}(\mathrm{G}_i|\mathrm{D})} \tag{B.1}$$

where D is the dataset and G_i represents one of the 543 CBNs over $\mathbf{V} = \{W_1, W_2, X, Y\}$.

Note also that the 543 CBNs can be partitioned into equivalence classes containing one or more CBNs. The 543 CBNs can be grouped into 185 equivalence classes (Gillispie & Perlman, 2002). Making use of this equivalence property it is possible to compute the sum score of all the 543 models (the denominator in the right hand side of Equation B.1) by scoring one representative model each from the equivalence class and multiplying by the corresponding number of models. The sum score can be computed using the following equation:

$$\sum_{i=1}^{543} \mathrm{Score}(\mathrm{G}_i|\mathrm{D}) \quad = \quad \sum_{j=1}^{185} \mathrm{Score}(\mathrm{E}_j|\mathrm{D}) * |(\mathrm{E_j})| \tag{B.2}$$

where D is the dataset, 185 is the total number of equivalence classes, E_j is a representative member of the jth equivalence class and $|(\mathrm{E}_j)|$ gives the number of CBNs in the jth equivalence class.

However, the current implementation of BLCD simply uses Equation B.1 to compute the sum score.

The reader is referred to Appendix D for theorems and proofs related to the MB procedure of BLCD that identifies the Markov blanket of a variable X from a dataset D, and to Appendix C for Y structure related theorems and proofs.

APPENDIX C

Y STRUCTURE THEOREMS

Definition 1 (Complete-table Bayesian network). A complete-table Bayesian network is one that contains all discrete variables and for which the probabilities that define the Bayesian network are described by contingency tables with no missing values.

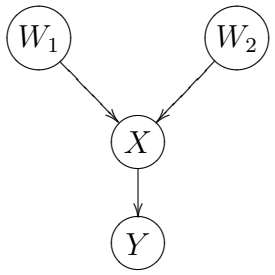
Definition 2 (Perfect map). A Bayesian network structure S is a perfect map of a distribution θ if each independence relationship in S (according to d-separation) is an independence relationship in θ, and each dependence relationship in S (according to d-connectivity) is a dependence relationship in θ.

Remark. Suppose Bayesian network B defines a joint distribution θ over all the variables in B. Let S be the structure of B. If the Markov and Faithfulness conditions hold for B, then S is a perfect map of θ.

Remark. In the results of this section, we will only be considering complete-table Bayesian networks that satisfy the Markov and Faithfulness conditions. We also assume positive distributions, that is, we only consider networks which do not contain probabilities of either 0 or 1.

Definition 3 (Independence equivalent). Two Bayesian network structures S and S^* are independence equivalent (Heckerman, 1995) if each independence relationship in S (according to d-separation) is an independence relationship in S^*, and each dependence relationship in S (according to d-connectivity) is a dependence relationship in S^*. Independence equivalence is also referred to as Markov equivalence.

Definition 4 (Y structure Bayesian network). A Y structure Bayesian network is a Bayesian network containing four variables that has the structure shown in Figure 49, where the node labels are arbitrary.



G_1

Figure 49: A Y structure.

Remark. We will use the following notation in regard to an arbitrary complete-table Bayesian network that satisfies the Markov and Faithfulness conditions and has a Y structure: B_y denotes the network, S_y denotes its structure, $\mathbf{V}_y$ denotes the four variables in the structure, Q_y denotes its complete table parameters, and θ_y denotes the correspondingly defined joint distribution over the four variables.

Lemma 1. *There is no other Bayesian network structure on the variables in* $\mathbf{V}_y$ *that is independence equivalent to* S_y.

Proof. We prove Lemma 1 by making use of the following theorem.

Theorem 1. *Two network structures* B_{s1} *and* B_{s2} *are independence equivalent iff they satisfy the following conditions (Verma & Pearl, 1991):*

1. B_{s1} *and* B_{s2} *have the same set of vertices.*
2. B_{s1} *and* B_{s2} *have the same set of edges ignoring arc directions.*
3. *If there is a configuration such as* $X \rightarrow Z \leftarrow Y$ *where* X *and* Y *are not adjacent ("V" structure) in* B_{s1}*, the same pattern is present in* B_{s2}*, and vice-versa.*

Remark. G_1 has a "V" structure $W_1 \rightarrow X \leftarrow W_2$. According to Theorem 1, Condition 3, to be in the independence equivalent class of G_1, a network structure G_2 should have the same "V" structure.

G_1 has a directed edge $X \rightarrow Y$ in addition to the "V" structure. According to Theorem 1, Condition 2, to be in the independence equivalent class of G_1, G_2 should have one of the following edges and none other apart from the "V" structure:

- $X \rightarrow Y$.
- $X \leftarrow Y$.

If G_2 has the edge $X \rightarrow Y$, $G_2 \equiv G_1$ (identical structure to G_1).

If G_2 has the edge $X \leftarrow Y$, two additional "V" structures $W_1 \rightarrow X \leftarrow Y$ and $W_2 \rightarrow X \leftarrow Y$ are created in G_2 violating Theorem 1, Condition 3.

This completes the proof of Lemma 1. □

Lemma 2. *Let* B *be a Bayesian network that contains the fewest number of parameters that can represent the population distribution. Let* B^* *be a Bayesian network that either cannot represent the population distribution, or can but does not contain the fewest number of parameters. Let* S *and* S^* *be the network structures of* B *and* B^**, respectively. Let* m *denote the number of iid cases* D *that have been sampled from the population distribution defined by* B.

$$\text{Then } \lim_{m\rightarrow\infty} \frac{P(S^*, D)}{P(S, D)} < 1. \tag{C.1}$$

Proof. The proof of Lemma 2 follows from the results in (Chickering, 2002), which in turn uses results in (Haughton, 1988). □

Theorem 2. *Let* B = (S,Q) *be a complete-table Bayesian network that contains* n *measured variables, where* S *and* Q *are the structure and parameters of* B*, respectively. Suppose that* B *defines a distribution* θ *on the* n *variables, such that* S *is a perfect map of* θ*. Let* D *be a database containing* m *complete cases on the* n *variables in* B*, for which the cases are iid samples from distribution* θ*. Let* B^* *be a Bayesian network with structure* S^* *that is not*

independence equivalent to B. *Suppose that* $P(\mathrm{S}, \mathrm{D})$ *and* $P(\mathrm{S}^*, \mathrm{D})$ *are computed using the BDe score with non-zero parameter and structure priors.*

$$\text{Then } \lim_{m \to \infty} \frac{P(\mathrm{S}^*, \mathrm{D})}{P(\mathrm{S}, \mathrm{D})} < 1. \tag{C.2}$$

Proof. If B^* cannot represent the generative distribution θ then according to Lemma 2 the current theorem holds. Suppose B^* can represent the generative distribution. Since by assumption B^* is not independence equivalent to B, B^* must contain all the dependence relationships in B, plus additional dependence relationships. Therefore B^* contains more parameters than B (Chickering, 2002, Proposition 8). Thus it follows from Lemma 2 that the theorem holds. □

Theorem 3. *Assume the notation and conditions in Theorem 2 and suppose the number of variables is four* ($n = 4$). *If* S *is the data generating structure and* S *is a Y structure, then in the large sample limit* $P(\mathrm{S}, \mathrm{D}) > P(\mathrm{S}^*, \mathrm{D})$ *for all* $\mathrm{S}^* \neq \mathrm{S}$. *Conversely, if* S *is the data generating structure and* S *is not equal to some Y structure,* S^*, *then in the large sample limit* $P(\mathrm{S}, \mathrm{D}) > P(\mathrm{S}^*, \mathrm{D})$.

Proof. The proof follows from Theorem 2 and Lemma 1. □

Remark. Theorem 3 shows (under the conditions assumed) that in the large sample a Y structure will have the highest BDe score if and only if it is the structure of the data generating Bayesian network.

Remark. Lemma 2 can be strengthened using results in (Nishii, 1988, Theorem 4) to show that the ratio is equal to 0, rather merely less than 1[1]. Correspondingly, Theorem 3 can be strengthened to state that the data generating structure has probability 1 and all other structures have probability 0[2]. This strengthened version of Theorem 3 implies that in the large sample limit, model averaging using Equation B.1 on page 157 of this appendix will derive an arc $X \to Y$ as causal and unconfounded with probability 1, if and only if it is a

[1] The results in (Nishii, 1988) are based on "almost surely" convergent proofs, which guarantee that in the large sample limit the data will with probability 1 support the stated convergence.

[2] If there are several Bayesian networks that contain the fewest number of parameters that can represent the data generating distribution, then the result states that the sum of their posterior probabilities is equal to 1.

causal and unconfounded arc within a Y structure of the data generating causal Bayesian network.

The proof of correctness of BLCD based on Y structure identification requires an understanding of the common properties of DAGs in the same Markov equivalence class even in the presence of hidden or latent variables. A class of structures that can represent the common properties of DAGs in the same Markov equivalence class are called partial ancestral graphs (PAGs). We next describe the theorems that make use of PAGs. A PAG has a richer representation compared to a DAG and makes use of the following types of edges.

1. $\rightarrow$
2. $\leftrightarrow$
3. $\circ\!\!\rightarrow$
4. $\circ\!\!-\!\!\circ$

Partial ancestral graphs

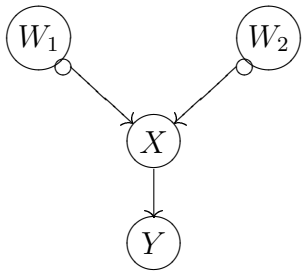
A Markov equivalence class of DAGs over a set of observed variables $\mathbf{O}$ is the set of all DAGs that contain at least the variables in $\mathbf{O}$ and that have the same set of d-separation relations among the variables in $\mathbf{O}$ (i.e. G_1 and G_2 are in the same Markov equivalence class over $\mathbf{O}$ if for all disjoint $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$, $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in G_1 iff $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in G_2). A PAG $\mathcal{P}$ over $\mathbf{O}$ is a graphical object with vertices $\mathbf{O}$ that represents the Markov equivalence class of DAGs $\mathcal{M}$ over $\mathbf{O}$ in two distinct ways:

1. A PAG represents the d-separation relations over $\mathbf{O}$ in $\mathcal{M}$.
2. A PAG represents the ancestor and non-ancestor relations among members of $\mathbf{O}$ common to every DAG in $\mathcal{M}$.

More specifically, it is possible to extend the concept of d-separation in a natural way to PAGs so that if PAG $\mathcal{P}$ represents the Markov equivalence class $\mathcal{M}$ over $\mathbf{O}$, then for all disjoint $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$, $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in $\mathcal{P}$ iff $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in every DAG in $\mathcal{M}$. A PAG is formally defined as stated below.

Definition 5 (PAG). The PAG $\mathcal{P}$ that represents a Markov equivalence class $\mathcal{M}$ over $\mathbf{O}$ can be formed in the following way:

1. X and Y are adjacent in $\mathcal{P}$ iff X and Y are d-connected conditional on every subset of $\mathbf{O}\backslash\{X, Y\}$.
2. If X and Y are adjacent in $\mathcal{P}$, there is an "$-$" (arrowtail) at the X end of the edge iff X is an ancestor of Y in every member of $\mathcal{M}$.
3. If X and Y are adjacent in $\mathcal{P}$, there is an "$>$" (arrowhead) at the Y end of the edge iff Y is not an ancestor of X in every member of $\mathcal{M}$.
4. If X and Y are adjacent in $\mathcal{P}$, an "o" at the X end of the edge between them places no constraint on the ancestor relations.



G_1

Figure 50: A Y PAG.

For example, suppose $\mathcal{M}$ is the Markov equivalence class of the Y structure. It can be shown that the PAG that represents a Y structure is in Figure 50, indicating that for every DAG in $\mathcal{M}$, the following conditions hold:

- X is not an ancestor of W_1 or W_2.
- Y is not an ancestor of X.
- X is an ancestor of Y.
- W_1 and W_2 may or may not be ancestors of X in different members of M.

Definition 6 (DAG PAG). For a PAG $\mathcal{P}$, if there is an assignment of arrowhead and arrowtails to the "o" endpoints in P such that the resulting DAG has the same d-separation relations as $\mathcal{P}$, then $\mathcal{P}$ is a DAG PAG.

For example, a Y PAG is a DAG PAG because the DAG in Figure 49 has the same d-separation relations as the PAG in Figure 50.

Remark. A DAG PAG can be parameterized in the same way as a corresponding DAG. Every DAG has the same d-separation over $\{W_1, W_2, X, Y\}$ as some PAG containing just $\{W_1, W_2, X, Y\}$.

The reader is referred to Section 3.1.2, (Spirtes et al., 1999) and (Spirtes et al., 2000, pages 299–301) for additional details about PAGs.

Definition 7 (Embedded pure Y structure). Let B be a causal Bayesian network with structure S. We say that B contains an *embedded pure Y structure* (EPYS) involving the variables W_1, W_2, X and Y, if the following conditions hold ($X<>Y$ means X and Y are d-connected, and $X><Y$ means that X and Y are d-separated):

1. $W_1><W_2$
2. $W_1<>X$
3. $W_2<>X$
4. $W_1<>W_2|X$
5. $X <>Y$
6. $W_1<>Y$
7. $W_2<>Y$
8. $W_1><Y|X$
9. $W_2><Y|X$

Context 1. Let B be a complete-table Bayesian network involving the variables W_1, W_2, X and Y. Furthermore, let B be the data generating model for data on just W_1, W_2, X and Y. In general, B may contain other variables, which we consider as hidden with regard to the data being generated on these four variables.

Let θ_{w1w2xy} be the data generating distribution on the variables W_1, W_2, X and Y that is given by a marginal distribution of B. Suppose that every d-separation condition among W_1, W_2, X and Y in B implies a corresponding independence according to θ_{w1w2xy} (call this the *marginal Markov condition*). Suppose also that every d-connection condition among

W_1, W_2, X and Y implies a corresponding dependence according to θ_{w1w2xy} (call this the *marginal faithfulness condition*).

To summarize:

1. Let S_y denote the Y structure in Figure 49.
2. Let $\mathbf{V}_{\mathrm{S}y}$ denote the variables in S_y.
3. Let B be the Bayesian network generating the data.
4. Let $\mathbf{V}_{\mathrm{B}}$ denote the variables in B.
5. In general $\mathbf{V}_{\mathrm{S}y} \subseteq \mathbf{V}_{\mathrm{B}}$.
6. Assume the marginal Markov and faithfulness conditions hold for B with respect to θ_{w1w2xy}.

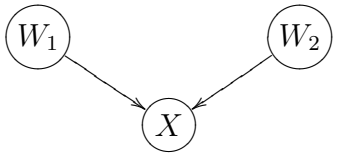


Figure 51: An unshielded collider X

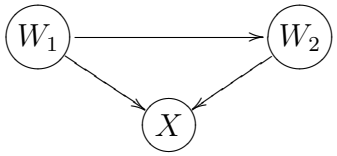


Figure 52: A shielded collider X

Definition 8 (Collider). A variable X is said to be a collider if it has two incoming arcs (arrowheads). If there is an arc from W_1 to X and an arc from W_2 to X, and W_1 and W_2 are not adjacent, X is said to be an *unshielded* collider (see Figure 51). In addition, if W_1 and W_2 are adjacent, X is said to be a *shielded* collider (see Figure 52).

Definition 9 (Non-collider). A variable X is said to be a non-collider if it does not have two incoming arcs (arrowheads). See Figure 53 and Figure 54 for examples.

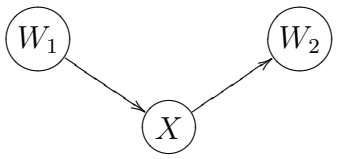


Figure 53: A non-collider X

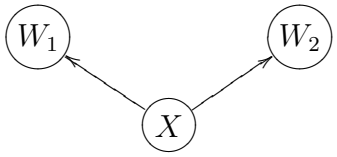


Figure 54: Another example of a non-collider X

Lemma 3. *If* B contains an EPYS, then for every DAG PAG $\mathcal{P}$ other than the Y PAG, either $\mathcal{P}$ contains more parameters than the Y PAG, or $\mathcal{P}$ contains a d-separation that is not in the EPYS.

Proof. Suppose that $\mathcal{P}$ does not contain a d-separation that is not in the EPYS, and hence does not contain a d-separation that is not in the Y PAG. It follows that $\mathcal{P}$ has a proper subset of the d-separation relations in the Y PAG. No PAG with the same adjacencies as the Y PAG, or that lacks an adjacency that is in the Y PAG has a proper subset of the d-separation relations in the Y PAG. Hence $\mathcal{P}$ has a proper superset of the adjacencies in the Y PAG. In addition if there is an unshielded collider (non-collider) in the Y PAG that is unshielded in $\mathcal{P}$, it is also a collider (non-collider) in $\mathcal{P}$. This entails that $\mathcal{P}$ represents a Markov equivalence class of DAGs that contains a DAG that is a proper supergraph of the Y DAG, and hence contains more parameters than the Y DAG (and the corresponding Y PAG). □

Lemma 4. *If* B *does not contain an EPYS, then either the Y PAG has a d-separation relation not in the d-separation relations over* **O** *of* B*, or there is a DAG PAG* $\mathcal{P}$ *over* **O**

such that every d-separation in $\mathcal{P}$ is in B*, and $\mathcal{P}$ contains fewer parameters than the Y PAG.*

Proof. First note that all of the d-connection and d-separation relations not explicitly mentioned in the definition of an EPYS can be shown to be entailed by the definition of an EPYS. Suppose that B does not contain an EPYS. Then either there is a d-separation relation in B among the variables in $\mathbf{O}$ that is not in the EPYS, or there is a d-separation relation in the EPYS that is not in B. If there is a d-separation relation in the EPYS that is not in B, then there is a d-separation relation in the Y PAG that is not in B. Suppose then that every d-separation relation in the EPYS is in B, but in addition there is a d-separation relation in B that is not in the EPYS. In other words B contains a proper superset of the d-separation relations in the EPYS, and hence in the Y PAG also. It can be shown that if B contains a proper superset of the d-separation relations over $\mathbf{O}$ in the Y PAG, then the d-separation relations in B over $\mathbf{O}$ can be represented by a PAG $\mathcal{P}$ that has a subset of the adjacencies of (and if the corresponding edges exist in $\mathcal{P}$, the same unshielded colliders and non-colliders as) the Y PAG. $\mathcal{P}$ represents a Markov equivalence class that contains a subgraph of the Y DAG, and hence has fewer parameters than the Y DAG (and corresponding Y PAG). □

Theorem 4. *Assume that Context 1 holds. In the large sample limit, in scoring DAGs on $\mathbf{V}_{\mathrm{S}y}$, BLCD assigns the highest score to* S_y*, iff* B *contains a corresponding EPYS, and if* B *contains such an EPYS, then X is an ancestor Y in* B*.*

Proof. If B does contain an EPYS, by Lemma 3, every DAG PAG $\mathcal{P}$ other than the Y PAG either contains more parameters than the Y PAG, or contains a d-separation not in the Y PAG. If $\mathcal{P}$ contains a d-separation not in the Y PAG and hence not in the EPYS, then $\mathcal{P}$ entails a conditional independence where by the faithfulness assumption there is a conditional dependence. Hence $\mathcal{P}$ cannot represent the population distribution.

It follows that either $\mathcal{P}$ cannot represent the distribution or contains more parameters than the Y PAG. By Lemma 2, in the large sample limit $\mathcal{P}$ gets a lower score than the Y PAG (and the corresponding Y DAG). In addition, because the Y PAG is the PAG that represents the Markov equivalence class over $\mathbf{O}$ ($= \{W_1, W_2, X, Y\}$) of all DAGs with the d-separation and d-connection relations of the EPYS, and there is a directed edge in the Y PAG from X to Y, there is a directed path from X to Y in B.

If B does not contain an EPYS, then by Lemma 4, either the Y PAG has a d-separation relation not in the d-separation relations of B over $\mathbf{O}$, or there is a DAG PAG $\mathcal{P}$ over $\mathbf{O}$ such that every d-separation in $\mathcal{P}$ is in B, and $\mathcal{P}$ contains fewer parameters than the Y PAG. In the former case, the Y PAG entails a conditional independence where by faithfulness the population distribution has a conditional dependence; in that case the Y PAG (and hence the Y DAG) cannot represent the marginal population distribution. In the latter case, because every d-separation in $\mathcal{P}$ is also in the d-separations over $\mathbf{O}$ of B, $\mathcal{P}$ can represent the marginal population distribution. So if B does not contain an EPYS, either the Y PAG does not represent the population distribution, or there is a DAG PAG $\mathcal{P}$ that can represent the population distribution that has fewer parameters than the Y PAG. By Lemma 2, in the large sample limit, $\mathcal{P}$ receives a higher score than the Y DAG (and the corresponding Y PAG). □

Remark. Theorem 4 indicates that local Bayesian causal discovery using Y structures is possible (under assumptions), even when the data generating process is assumed to be a causal Bayesian network with hidden variables.

APPENDIX D

MARKOV BLANKET THEOREMS

Theorem 5. *The MB procedure of BLCD finds the Markov blanket of a node X under the assumptions of Markov, faithfulness and large sample size.*

Remark. The MB procedure (MBP) uses a greedy forward and backward search in seeking the Markov blanket of X, which we denote MB(X). The set MB(X) is the Markov blanket of X in a data generating network B. Let **R** denote all the variables in B.

Let **H** be a set that represents a *putative* Markov blanket of X. MBP uses BDe to score how well **H** predicts X. MBP's direct goal is to find among all the nodes in **R** the smallest set **H** that predicts X as well as possible. In predicting X, set **H** is viewed by the BDe scoring procedure as the parents of X. However, note that **H** actually represents a putative Markov blanket of X.

In what follows, we use three steps to show that in the large sample limit, MBP finds the Markov blanket of X, assuming that the Markov and faithfulness conditions hold. We first describe the components of the BDe score, P(S,D) (where S is the Bayesian network structure and D is the dataset), that will be useful in analyzing the behavior of MBP. Next, we show how the forward stepping search of MBP finds a superset of the Markov blanket of X. Finally, we show how the backward search step reduces that superset to just the nodes in the Markov blanket of X in B.

D.1 PROOF

D.1.1 The components of the score

The log of P(S,D) is called the *Bayesian scoring criterion* (Chickering & Meek, 2002), which can be expressed as follows:

$$log(\mathrm{P(S,D)}) = log(\mathrm{P(S)}) + log(\mathrm{P(D|S)}) \tag{D.1}$$

Geiger et al. show that a complete-table Bayesian network is a curved exponential model (Geiger et al., 2001). As discussed in (Chickering & Meek, 2002), Haughton derives the following form for the Bayesian scoring criterion for curved exponential models (Haughton, 1988):

$$log(\mathrm{P(S,D)}) = log(\mathrm{P(D}|\hat{\theta}_{\mathrm{S}})) - \frac{d}{2}log(m) + O(1) \tag{D.2}$$

where $\hat{\theta}_{\mathrm{S}}$ denotes the maximum likelihood distribution as defined by the values of the model parameters, d is the dimension of the model, m is the number of records in D, and $O(1)$ is some constant that does not depend on m. Haughton shows that the first term in Equation D.2 is $O(m)$. Clearly the second term in that equation is $O(log(m))$ and the third term is $O(1)$. Thus, as $m \to \infty$ the first term dominates the second and third terms, and the second dominates the third.

D.1.2 Forward search

Since the likelihood term in Equation D.2 dominates the other two terms, the MB procedure will continue adding nodes to $\mathbf{H}$ as long as they increase the likelihood. Suppose $Z \ni \mathbf{H}$, and in B node X is d-connected to Z given $\mathbf{H}$. According to the faithfulness condition, X is dependent on Z given $\mathbf{H}$. Thus, adding Z to $\mathbf{H}$ will increase the likelihood term in the large sample limit. The forward step of the MB search procedure will therefore add Z to $\mathbf{H}$. Such node additions will continue until there is no node Z such that $Z \ni \mathbf{H}$ but X is d-connected to Z given $\mathbf{H}$. At that point, $\mathbf{H}$ includes the MB(X) in B by the definition of a Markov blanket.

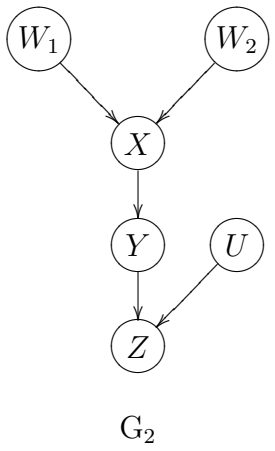


Figure 55: A CBN with five nodes to illustrate the MB procedure.

D.1.3 Backward search

Upon termination of the forward search, there may be a node Z in $\mathbf{H}$ that is independent of X given $\mathbf{H} \setminus \{\mathrm{Z}\}$. For example, if the causal Bayesian network G_2 shown in Figure 55 is the data generating structure, the set $\mathbf{H}$ could contain nodes Z, U, Y, W_1, W_2, in that order, after the forward search. Clearly, here Z is independent of X given $\mathbf{H} \setminus \{\mathrm{Z}\}$. Removing Z from H will increase the score given by Equation D.2 in the large sample limit, because d in the second term of the equation will decrease while the first term will remain the same or go down by a quantity smaller than the increase in score obtained from the second term. The backward search of MBP sequentially removes all such nodes from $\mathbf{H}$, leaving only those nodes in $\mathbf{H}$ that together maximize the first term in Equation D.2 while minimizing the second term. Note that the backward search will not remove a node W in $\mathbf{H}$ that is not independent of X given $\mathbf{H} \setminus \{\mathrm{W}\}$, because, removing such a W will decrease the likelihood term. The nodes that remain in $\mathbf{H}$ are the minimum set required to render X independent of the nodes in the complement of $\mathbf{H} \cup \{\mathrm{X}\}$. Thus, $\mathbf{H}$ satisfies the criteria for being the Markov blanket of X.

This completes the proof of the MB procedure. □

BIBLIOGRAPHY

Abramson, B., Brown, J., Edwards, W., Murphy, A., & Winkler, R. L. (1996). Hailfinder: A Bayesian System for Forecasting Severe Weather. *International Journal of Forecasting*, *12*, 57–71.

Aliferis, C., & Cooper, G. (1994). An evaluation of an algorithm for inductive learning of Bayesian belief networks using simulated datasets. *Proceedings of the conference on Uncertainty in Artificial Intelligence* (pp. 8–14). San Francisco: Morgan Kaufmann.

Aliferis, C. F., Tsamardinos, I., & Stanikov, A. (2003). HITON, A novel markov blanket algorithm for optimal variable selection. *Proceedings of the AMIA Fall Symposium*.

Andreassen, S., Woldbye, M., Falck, B., & Andersen, S. K. (1987). MUNIN — A causal probabilistic network for interpretation of electromyographic findings. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 366–372). San Mateo, CA: Morgan Kaufmann.

Armitage, P., Berry, G., & Matthews, J. (2002). *Statistical Methods in Medical Research*. Oxford: Blackwell Science. 4 edition.

Beinlich, I. A., Suermondt, H., Chavez, R. M., & Cooper, G. F. (1990). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Proceedings of the Second European Conference on Artificial Intelligence in Medicine* (pp. 247–256). London: Chapman and Hall.

Benson, K., & Hartz, A. J. (2000). A comparison of observational studies and randomized controlled trials. *The New England Journal of Medicine*, *342*, 1878–86.

Bowden, R., & Turkington, D. (1984). *Instrumental Variables*. Camdridge, U.K: Cambridge University Press.

Buntine, W. (1991). Theory refinement on bayesian networks. *Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)* (pp. 52–60). San Mateo, CA: Morgan Kaufmann Publishers.

Cheng, J., Bell, D. A., & Liu, W. (1997). An algorithm for Bayesian belief network construction from data. *Proceedings of the 6th international workshop on Artificial Intelligence and Statistics*.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, *49*, 327–35.

Chickering, D. (2002). Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research*, *3*, 507–554.

Chickering, D., & Meek, C. (2002). Finding optimal bayesian networks. *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence* (pp. 94–102). Edmonton, AB: Morgan Kaufmann.

Chickering, D., & Pearl, J. (1996). A clinician's tool for analyzing non-compliance. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96),* Portland, OR (pp. 1269–1276).

Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. In D. Lenz and H. Fisher (Eds.), *Learning from Data*, 121–130. Springer-Verlag.

Chickering, D. M., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In D. Geiger and P. Shenoy (Eds.), *Proceedings of the 13th annual conference on Uncertainty in Artificial Intelligence*, 80–89. San Francisco, CA: Morgan Kaufmann.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462–467.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*, 309–347.

Cooper, G. F. (1997). A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, *1*, 203–224.

Cooper, G. F. (1999). An Overview of the Representation and Discovery of Causal Relationships Using Bayesian Networks. In C. Glymour and G. F. Cooper (Eds.), *Computation, Causation, and Discovery*, 3–62. Cambridge, MA: MIT Press.

Cooper, G. F., & Yoo, C. (1999). Causal discovery from a mixture of experimental and observational data. *Uncertainty in Artificial Intelligence 99* (pp. 116–125). San Francisco, California: Morgan Kaufmann Publishers.

Daniel, W. W. (1991). *Biostatistics: A foundation for analysis in the health sciences.* John Wiley and Sons, Inc. 5 edition.

Dash, D., & Druzdzel, M. J. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. *Uncertainty in Artificial Intelligence 99* (pp. 142–149). San Francisco, California: Morgan Kaufmann Publishers.

Dempster, A., Laird, N., & D, R. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, *B39*, 1–38.

Edwards, W. (1998). Hailfinder—Tools for and Experiences with Bayesian Normative Modeling. *American Psychologist*, *53*, 416–428.

Friedman, C., & Wyatt, J. (1997). *Evaluation Methods in Medical Informatics*. Springer.

Friedman, N. (2004). Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, *303*, 799–805.

Friedman, N., & Goldszmidt, M. (1999). Learning Bayesian networks with local structure. In M. I. Jordan (Ed.), *Learning in graphical models*, 421–459. Cambridge, MA: MIT Press.

Friedman, N., & Koller, D. (2000). Being Bayesian about network structure. *Uncertainty in Artificial Intelligence 2000* (pp. 201–210). San Francisco, California: Morgan Kaufmann Publishers.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.

Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The "Sparse Candidate" algorithm. *Uncertainty in Artificial Intelligence 99* (pp. 206–215). San Francisco, California: Morgan Kaufmann Publishers.

Geiger, D., Heckerman, D., King, H., & Meek, C. (2001). Stratified exponential families: Graphical models and model selection. *Annals of Statistics*, *29*, 505–529.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–742.

Gillispie, S. B., & Perlman, M. D. (2002). The size distribution for Markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, *141*, 137–155.

Glymour, C., & Cooper, G. F. (Eds.). (1999). *Computation, Causation, and Discovery*. Cambridge, MA: MIT Press.

Glymour, C., Spirtes, P., & Richardson, T. (1999). On the possibility of inferring causation from association without background knowledge. In C. Glymour and G. F. Cooper (Eds.), *Computation, Causation, and Discovery*, 323–331. Cambridge, MA: MIT Press.

Haughton, D. (1988). On the choice of a model to fit data from an exponential family. *Annals of Statistics*, *16*, 342–355.

Heckerman, D. (1995). A bayesian approach to learning causal networks. *Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 285–295). San Francisco, California: Morgan Kaufmann Publishers.

Heckerman, D. (1996). *A tutorial on learning with bayesian networks* (Technical Report). Microsoft Research Technical Report MSR-TR-95-06. Updated Nov. 1996.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197–243.

Heckerman, D., Horvitz, E., & Nathwani, B. (1992). Towards normative expert systems: Part I The Pathfinder Project. *Methods of Information in Medicine*, *31*, 90–105.

Heckerman, D., Meek, C., & Cooper, G. (1999). A bayesian approach to causal discovery. In C. Glymour and G. F. Cooper (Eds.), *Computation, Causation, and Discovery*, 141–165. Cambridge, MA: MIT Press.

Heckerman, D., & Nathwani, B. (1992). Towards normative expert systems: Part II probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, *31*, 106–116.

Henrion, M. (1986). Propagating uncertainty in bayesian networks by probabilistic logic sampling. *Proceedings of the 2nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-86)* (pp. 0–0). New York, NY: Elsevier Science Publishing Company, Inc.

Herbert, N. (1985). *Quantum Reality*. Garden City, N.Y: Anchor.

Ioannidis, J. P., Haidich, A.-B., Pappa, M., Pantazis, N., Kokori, S. I., Tektonidou, M. G., Contopoulos-Ioannidis, D. G., & Lau, J. (2001). Comparison of evidence of treatment effects in randomized and nonrandomized studies. *JAMA*, *286*, 821–830.

Kleinbaum, D. (1998). *Applied Regression Analysis and Other Multivariable Methods*. Duxbury Press. 3 edition.

Kristensen, K., & Rasmussen, I. (2002). The use of a Bayesian network in the design of a decision support system for growing malting barley without use of pesticides. *Computers and Electronics in Agriculture*, *33*, 197–217.

Lam, W., & Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, *10*, 269–294.

Lander, E. S. (1999). Array of hope. *nature genetics supplement*, *21*, 3–4.

Mani, S. (2000). Finding higher order (multivariate) causal influences using a Bayesian network approach. Term Project for Decision Analysis Course (unpublished).

Mani, S., & Cooper, G. F. (1999). A study in causal discovery from population-based infant birth and death records. *Proceedings of the AMIA Annual Fall Symposium* (pp. 315–19). Philadelphia, PA: Hanley & Belfus.

Mani, S., & Cooper, G. F. (2000). Causal discovery from medical textual data. *Proceedings of the AMIA Annual Fall Symposium* (pp. 542–546). Philadelphia, PA: Hanley & Belfus.

Mani, S., & Cooper, G. F. (2001). A simulation study of three related causal data mining algorithms. *International Workshop on Artificial Intelligence and Statistics* (pp. 73–80). San Francisco, California: Morgan Kaufmann Publishers.

Margaritis, D., & Thrun, S. (2000). Bayesian network induction via local neighborhoods. *Advances in neural information processing systems* (pp. 505–511). Cambridge, MA: MIT Press.

Meek, C. (1995). Strong Completeness and Faithfulness in Bayesian Networks. *Proceedings of the Conference on Uncertainty in Artificial Intelligence* (pp. 411–418). San Francisco, California: Morgan Kaufmann.

Miller, R. (1981). *Simultaneous Statistical Inference*. Springer-Verlag. 2 edition.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Moore, A., & Wong, W.-K. (2003). Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning. *Proceedings of the 20th International Conference on Machine Learning (ICML '03)* (pp. 552–559). Menlo Park, California: AAAI Press.

National Center for Health Statistics (1996). *1991 Birth Cohort Linked Birth/Infant Death Data Set*. National Center for Health Statistics. CD-ROM Series 20—No. 7.

Neapolitan, R. (1990). *Probabilistic reasoning in expert systems: Theory and algorithms*. New York: John Wiley and Sons.

Nishii, R. (1988). Maximum likelihood principle and model selection when the true model is unspecified. *Journal of Multivariate Analysis*, *27*, 392–403.

Pearl, J. (1991). *Probabilistic Reasoning in Intelligent Systems*. San Francisco, California: Morgan Kaufmann. 2 edition.

Pearl, J. (1994). *Mediating Instrumental Variables* (Technical Report). UCLA.

Pearl, J. (1995). On the Testability of Causal Models with Latent and Instrumental Variables. *Proceedings of the Eighteenth Conference of Uncertainty in Artificial Intelligence* (pp. 435–443). San Francisco, CA: Morgan Kaufmann.

Pe'er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 1–9.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., & Nolan, G. (2005). Causal protein-signalling networks derived from multiparameter single-cell data. *Science*, *308*, 523–529.

Salmon, W. C. (1997). *Causality and Explanation*. New York: Oxford University Press.

Silverstein, C., Brin, S., Motwani, R., & Ullman, J. (2000). Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, *4*, 163–192.

Singh, M., & Valtorta, M. (1995). Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, *12*, 111–131.

Spirtes, P. (2001). An anytime algorithm for causal inference. *International Workshop on Artificial Intelligence and Statistics* (pp. 121–128). San Francisco, California: Morgan Kaufmann Publishers.

Spirtes, P., & Cooper, G. (1999). An experiment in causal discovery using a pneumonia database. *Artificial Intelligence and Statistics 99* (pp. 162–168). San Francisco, California: Morgan Kaufmann Publishers.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, Prediction, and Search*. New York: Springer-Verlag.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, and Search*. Cambridge, MA: MIT Press. 2 edition.

Spirtes, P., & Meek, C. (1995). Learning bayesian networks with discrete variables from data. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 294–299). Menlo Park, California: AAAI Press.

Spirtes, P., Meek, C., & Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. In C. Glymour and G. F. Cooper (Eds.), *Computation, Causation, and Discovery*, 211–252. Cambridge, MA: MIT Press.

Suojanen, M., Olesen, K. G., & Andreassen, S. (1997). A method for diagnosing in large medical expert systems based on causal probabilistic networks. *Lecture Notes in Artificial Intelligence: Artificial Intelligence in Medicine, AIME97* (pp. 285–295). Springer.

Tian, J., & Pearl, J. (2002). On the Testable Implications of Causal Models with Hidden Variables. *Proceedings of the Eighteenth Conference of Uncertainty in Artificial Intelligence* (pp. 519–527). San Francisco, CA: Morgan Kaufmann.

Tsamardinos, I., Aliferis, C. F., & Stanikov, A. (2003). Time and sample efficient discovery of markov blankets and direct causal relations. *Proceedings of the 9th CAN SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 673–678).

Verma, T., & Pearl, J. (1991). *Equivalence and synthesis of causal models* (Technical Report). Cognitive Systems Laboratory, University of California at Los Angeles.

Wallace, C. S., & Korb, K. B. (1999). Learning linear causal models by MML sampling. In A. Gammerman (Ed.), *Causal models and intelligent data management*, 89–111. New York: Springer.

INDEX