12-1-2010

Three algorithms for causal learning

Roshan Ram Rammohan

Roshan Ram Rammohan

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

______________________________, Chairperson

____

____

____

____

____

____

____

____

Three Algorithms for Causal Learning

by

Roshan Ram Rammohan

B.E., Bangalore University, 1999
M.S., Dept. of Computer Science, University of New Mexico, 2006

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2010

Dedication

To my parents,
Sengunam Vishwanathan Rammohan and Radha Rammohan,
and my brother Rohant Shyam
for their limitless love and unflinching support.

Acknowledgments

Graduating with a Bachelor's degree in instrumentation and electronics engineering, I had no formal background in the computational or cognitive sciences. However, I was an avid reader of science fiction and I had nurtured a strong conviction that intelligence is not qualitatively different from computation. Following a brief stint as a researcher helping computational neuroscientists study the neural responses in mice, I decided to pursue graduate studies to understand intelligence from a computational perspective.

I wanted to find out if my conviction was true, and if so, how far computational intelligence is practical. Given this, I am extremely thankful to my good fortune and circumstance that I was able to find the best possible dissertation advisor, Professor George F. Luger. George is not only extremely knowledgable and concerned with the practical aspects of AI and related technology, but also indulges in deep analytic thought over the bigger philosophical questions about intelligence, a rare combination among researchers. I am forever indebted and grateful to George for listening patiently to all my ideas and struggles as a graduate student, the constant motivation and support, the invaluable scientific critique, all the great writing tips and most of all, for being a very good friend. This dissertation would not have been possible without his guidance. Thank you George!

I am also indebted to the members of my dissertation committee, for providing me with feedback, appreciation and constructive suggestions throughout the development of this research. I would like to thank Professor Tom Caudell, for the numerous interesting discussions that have piqued my intellect and for his unshaking confidence in my abilities as a researcher. I would like to thank Professor Lance Williams, for his keen insights and advice on several technical aspects of the dissertation and for his patient constructive criticism on writing style. I am also indebted to Dr. Carl Stern, for making sure that I am addressing all the important and relevant questions. I am specially grateful to Professor Mahmoud Reda Taha, for his keen interest in my academic progress over the years, for introducing me to several application areas and for his generous financial support. I whole heartedly thank you all!

Finally, I would like to thank my family and good friends for helping me survive this endeavor by keeping me sane, healthy, happy and loved. Лидия, моя лубов, вы на каждой странице этой работы и мое сердце.

Three Algorithms for Causal Learning

by

Roshan Ram Rammohan

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2010

Three Algorithms for Causal Learning

by

Roshan Ram Rammohan

B.E., Bangalore University, 1999
M.S., Dept. of Computer Science, University of New Mexico, 2006
Ph.D., Computer Science, University of New Mexico, 2010

Abstract

The field of causal learning has grown in the past decade, establishing itself as a major focus in artificial intelligence research. Traditionally, approaches to causal learning are split into two areas. One area involves the learning of structures from observational data alone and the second, involves the methodologies of conducting and learning from experiments. In this dissertation, I investigate three different aspects of causal learning, all of which are based on the causal Bayesian network framework. Constraint based structure search algorithms that learn partially directed acyclic graphs as causal models from observational data rely on the faithfulness assumption, which is often violated due to inaccurate statistical tests on finite datasets. My first contribution is a modification of the traditional approaches to achieving greater robustness in the light of these faults. Secondly, I present a new algorithm to infer the parent set of a variable when a specific type of experiment called a 'hard intervention' is performed. I also present an auxiliary result of this effort, a fast algorithm to estimate the Kullback Leibler divergence of high dimensional distributions from datasets. Thirdly, I introduce a fast heuristic algorithm to optimize the number

and sequence of experiments required towards complete causal discovery for different classes of causal graphs and provide suggestions to implementing an interactive version. Finally, I provide numerical simulation results for each algorithm discussed and present some directions for future research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"*A physical symbol system has the necessary and sufficient means for general intelligent action.*" - Alan Newell and Herbert Simon [NS76]

Since the beginnings of enquiry and intelligent thought, humans have tried to find explanations for the world around them. Observations about the natural world were often regarded as *effects* of some *causal* entity, the actions of fantastic and powerful god(s) or some other natural entity. Today, our understanding of the natural world is much better developed than in primitive times, and an increasing amount of explanations for observations are based on well documented and tested cause and effect models. However, the search for causal knowledge seems to be growing at an increasing rate. With new technologies developed by enhanced knowledge, we are now able to observe several new phenomena in much greater detail, thereby increasing the need for more explanations of these phenomena. The growth in the fields of electronics, computing, sensor technologies, imaging, etc. have proven to be great accelerants to fields like biology, medical sciences, geology and meteorology, etc., where previously data was scarce and hard to obtain. Today, these fields have become data-rich and what has become sparse (in relative terms) is the availability of skillful human resources to analyze this data and gather useful insights. One way to bridge this increasing gap between data and the limited abilities of human faculties, including memory and reasoning mechanisms, is by building computational augmentations of our native cognitive abilities.

In his *Posterior Analytics*, Aristotle wrote, "... we have scientific knowledge when we know the cause..." [Aric]. The practice of explaining observed phenomena is the overall objective of all scientific inquiry. The multitude of human sensory abilities: vision, hearing, touch, etc., extended by the increasing sophistication of tools and measuring devices has enabled us to access and manipulate our environment in novel ways. To systematically construct causal knowledge out of the environment we observe, modern scientists around the globe follow a set of guidelines in the recording of observations, building hypotheses and validating them with controlled and repeatable tests. This has been termed *the scientific method* and one of its most popular forms is known as the *hypothetico-deductive model* of scientific research [Whe47].

An example of an algorithmic statement of the hypothetico-deductive method outlined in Peter Godfrey-Smith's "Theory and Reality" [GS03] is as follows.

1. Gather data (observations about something that is unknown, unexplained, or new).
2. Hypothesize an explanation for the data.
3. Deduce a consequence of that explanation (a prediction) and formulate an experiment to see if the predicted consequence is observed.
4. Wait for corroboration. If there is corroboration, go to step 3. If not, the hypothesis is falsified. Go to step 2.

Progress in the field of artificial intelligence (AI) has assisted in the development of computer algorithms that replicate and augment several specific human cognitive abilities. Towards this goal, most AI systems follow paradigms that are combinations of several tasks that each mimic some abstract component of human cognition. These include but are not restricted to: receiving sensory input [Dav04, BG95]; interpreting it in a format susceptible to the platform of computation (feature extraction) [Web02]; distinguishing patterns (mining) [RCKW05]; developing a concise representation of it in memory (learning) [Mit97]; and retrieving this stored information for future tasks (inference and prediction) [Pea88]. These developments provide the essential tools and set the stage for the next big step in AI.

The pursuit of science and the ability to discern causality is one of the most advanced and unique cognitive ability of our species. It has been instrumental in our progress from primitive hunter-gatherers roaming the wild to our collective self-view as civilized and 'sentient' beings. Replicating this ability in AI would be a great accelerant to the progress of science and humanity. To infuse the spirit and rigor of scientific inquiry into AI systems, the automation of casual learning is a key goal and an important milestone towards the creation of an "artificial scientist".

As an example to how machines can contribute to scientific knowledge, we introduce a very interesting recent development. The advances in robotics and the streamlining of biological testing methodologies support the creation of robotic scientists [SAB+10, KRO+09]. *Adam* and *Eve* are two large integrated robotic systems which advance the automation of both the hypothetico-deductive model and the recording of experiments in sufficient detail to enable reproducibility. *Adam* is reported to have autonomously generated functional genomics hypotheses about the yeast *Saccharomyces cerevisiae* and experimentally tested these hypotheses by using laboratory automation[SAB+10]. The methodology for generating hypotheses is driven by inductive logic programming that is guided by heuristics set in place according to the expert opinions of biologists. The conclusions of these robotic scientists have been verified through manual experiments. The central hope of this dissertation is that these robotic scientists, aptly named *Adam* and *Eve*, and their successors, equipped with more powerful learning algorithms, take a proverbial bite from the apple of causal knowledge to usher in an exciting and promising era for artificial intelligence, and science in general.

1.1 "True" AI and Causality

The construction of knowledge often begins with the observation of a correlation (covariance) between events or quantities. To make an assertion on the causal mechanism related to a covariance, we must also be able to discern the direction of this influencing mechanism. This usually involves a more careful examination of these trends, under multiple conditions, often supplemented by laboratory tests.

Analogous to the problem of acquiring causal knowledge is the problem of recording it with a suitable representation. Many AI systems represent knowledge without explicitly stating the direction of influence, perhaps because they have no particular method of discerning it in the first place. For example, consider an artificial Neural Network (ANN), a very popular tool in AI systems used as function approximators and non-linear discriminators. An ANN when trained on a suitable set of sample inputs using the "back propagation" algorithm can successfully provide a 'black box' approximation of a function with n inputs and m outputs [Hay94].

$$Y = f(X), \text{where } X = \{x_1 \dots x_n\}, Y = \{y_1 \dots x_m\} \tag{1.1}$$

The direction of influence which is most common when we think of functions is that inputs "cause" outputs. We understand and often speak of the volume of traffic on the roads as a function of the day of the week, not the other way around. However, it might be convenient to do just that, when we are interested in making a backward inference about the day of the week, given that we observe a certain traffic volume. In probabilistic models, such a semantic reversal is accomplished by an application of the Bayes' rule [BP63]. ANN's can help us do that as well, as directionality is entirely artificial and and its interpretation is external to its capabilities. Assuming sufficient data, one could reverse the input-output semantics and obtain a neural network that approximates $X =$

$f^{-1}(Y)$, just as easily. While such semantic inversions are helpful in some applications, in other applications it is desirable to identify the directions of causality unambiguously so that it is possible to explicitly represent it. Therefore, a desirable feature of a knowledge representation for causality is to make explicit the directionality of relationships.

Knowledge is more meaningful when it is represented by causal models, as opposed to associative models. It provides us with valuable intuition about the mechanism of the underlying system. For example, the following truth table representing the relationship between three Boolean variables A,B and C is an associative model. It does not provide insight into how this relationship has come to be, in other words, 'how does it work'? The AND-gate with independent inputs A,B and output $C = A \wedge B$, however, is a causal model that has much better explanatory power.

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

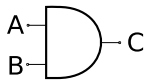


Figure 1.1: A Truth Table for 3 Boolean variables, A, B and C and a corresponding AND-gate causal model.

1.2 Causal Models in Artificial Intelligence

The two most popular forms of causal models are

1. Structural Equations
2. Causal Bayesian Networks

Structural equations first appeared in studies in the fields of genetics and economics [Wri21, Haa43, Sim53]. In its most general form, a structural equation is a functional

causal model of the form

$$x_i = f_i(pa_i, u_i), \qquad i = 1, \ldots, n \tag{1.2}$$

where pa_i denotes the "parents" or the set of variables that are the immediate causes of X_i, and U_i represents the errors in each equation due to omitted factors[HP01]. Eq (1.2) is a non-linear generalization of the linear Structural Equation Model (SEM) which is very popular in econometrics and the social sciences [Haa43, Sim53]:

$$x_i = \sum_{k \neq i} \alpha_{ik} x_k + u_i, \qquad i = 1, \ldots, n \tag{1.3}$$

In the linear model the set of parents, pa_i, is simply represented by the variables on the r.h.s. that have non-zero coefficients. Each equation is a *structural causal model* representing an autonomous mechanism which determines the value of a single variable on the *l.h.s.* The difference between structural equations and algebraic linear equations is that while the latter are characterized by solutions to the entire set of equations the former are characterized by solutions for each individual equation. This means that each individual equation in itself represents a valid model of reality. Functional causal models can also be visualized as a graphical representation of causality where each variable is a node in the graph, where there is an arc directed towards the l.h.s. variable's node from every other variable that has a nonzero coefficient in the r.h.s. of each structural equation. When we have multiple variables of interest, it is also useful to think of dependencies among then to be represented by a directed acyclic graph (DAG), where the arcs of the graph represent similar functional dependencies like in SEMs (sources *cause* targets). *Causal Bayesian networks* introduced by Pearl in [Pea88] is one such representation. It interprets causal structure as a DAG whose nodes are the variables or the quantities of interest and every direct causal link between two quantities is represented as a directed arc.

Technically, the absence of an arc between two nodes implies that the two nodes are conditionally independent, i.e., given some background information other than these nodes, namely the values of another disjoint set of nodes, one cannot gain any extra information about one node from the other. Moreover, Bayesian networks model the relationships among the nodes probabilistically rather than functionally. They encode the joint probability distribution among these variables as a factorization of a set of conditional probability distributions along a particular ordering of the variables. The DAG-based causal representations in Bayesian networks also tend to be human-readable and coincide with intuition. It has been suggested that humans themselves represent causal knowledge in their brain in abstractions similar to Bayesian networks [GGS$^+$04]. We will focus on causal Bayesian networks as our primary experimental framework in this thesis and will introduce them in greater detail in chapter 2.

Some of the early work on Bayesian networks made the assumption that the causal structure was given as prior knowledge elicited from experts and the designers of such systems [Pea88]. Later on, researchers started focusing on methods to learn dependencies among the quantities from data automatically (structure learning systems) [HGC95, KD05, RD06, CL68, CH92]. While many of these methods have proven successful in terms of detecting and recording conditional independence relations, coincidence with the actual causal structure is either by chance or is an artifact of some well designed and human conditioned heuristics.

In the past decade a lot of research has focused on the automated learning of causal structures from data. The area has attracted interest from a number of fields, ranging from economics, bio-informatics, to artificial intelligence [Pea00, Rub06, ROR07]. Judea Pearl and his colleagues have been the most important contributors to the field, and have proposed a refreshing, formal, and thorough treatment of this topic paving the way for very promising future research [Pea00, SGS00, TP01a, ES06].

Tr/Dow	1	2	3	4	5	6	7
high	n_{1h}	n_{2h}	n_{3h}	n_{4h}	n_{5h}	n_{6h}	n_{7h}
low	n_{1l}	n_{2l}	n_{3l}	n_{4l}	n_{5l}	n_{6l}	n_{7l}

Table 1.1: A contingency table between traffic Tr and day of the week DoW

1.3 Causal Learning

The first step towards discerning causal structure is to discern the structures of covariance. Let us briefly step back to our example regarding traffic and the day of the week mentioned in Section 1.1. Let us assume that the variable Tr represents the total traffic volume on a particular road in a city and it can take one of two values $high, low$, and DoW represents the day of the week and takes values from $[1 - 7]$. Suppose we record observations on a large sample of instances and measure the frequencies of $high$ and low vs. DoW in a *contingency table* [Pea04] as shown in Table 1.3. In the table, n_{ij} represents the count for observations where DoW is i and Tr is j.

We can use one of several possible statistical tests to determine if there is an association between these quantities. One such test which is widely used is Pearson's *chi-square* (χ^2) test for unconditional independence [Pea04]. The test is based on computing the χ^2 test statistic based on a two dimensional contingency table with r rows and c columns (see above), and rejecting the null hypothesis that the two events are independent based on its value. With the hypothesis of independence, we can calculate the theoretically expected values of each cell in the contingency table as the normalized product of the marginals for a sample size of N as follows:

$$E_{ij} = \frac{\left(\sum_{k=1}^{c} n_{ik}\right)\left(\sum_{k=1}^{r} n_{kj}\right)}{N}. \tag{1.4}$$

The value of the χ^2 test statistic is

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - E_{ij})^2}{E_{ij}} \,. \tag{1.5}$$

A widely accepted practice is to reject the null hypothesis of independence if the χ^2 probability for $(r-1)(c-1)$ degrees of freedom is less than or equal to a pre-determined significance level (typically 0.05). Several other tests exist for other kinds of applications and some (like the G^2 test and conditional cross entropy) can check for conditional independence given observations that include a third variable [FW95]. With multiple variables we can now carry out a series of pairwise tests among these variables, and whenever we reject independence, conditional or otherwise, we can include an undirected arc in our causal model. The undirected arc represents the fact that we observe a covariance or dependency, but are yet unaware of its direction of influence (also termed as *orientation*). For example, we observe that the values of $Tr = high$ always coincides with the values for $2 \leq DoW \leq 6$ an $Tr = low$ always coincides with the weekend ($DoW = 1$ or 7).

It has been shown that under assumptions of minimality, some of the orientations can be determined from clues obtained from independence tests when more than two variables are involved [VP91]. The conditioning variable in an independence test can be viewed as a control variable and the results of these tests can help determine whether certain causal links remain persistent across all possible models. In such a case, these links are "stable" and must exist in the causal model. We will delve into the technical details of this method and discuss it in more detail in Section 2.2. However, not all orientations can be determined using the previous method. In scientific experimentation certain "control variables" are set to predetermined values and observations are recorded under these conditions. This idea has been formalized into the theory of causal learning as the "calculus of interventions" [Pea00]. Suppose the city in our previous example builds a well designed and popular underground metro-rail system effectively shifting the distibution on Tr towards low and we record observations under this condition. Suppose we observe that there is no change in the marginal frequencies of DoW than we did previously, namely $\frac{1}{7}$ each, we can conclude

that Tr does not have any causal effect on DoW (as expected). On the other hand we can modify the nature of DoW. The town adds an extra day, Friday, to the weekend holiday. If we now record a change in the marginal distribution on Tr, which would now presumably reflect the drop in Friday traffic, we can then conclude that DoW does indeed have a causal effect on Tr. Note that interventions on a variable have the potential to change mechanisms that are influenced by that variable. Interventions are different and should not be confused with the concept of conditioning in probability theory. Conditioning merely allows us to observe different projections of the the existing distribution and has no ability to modify the underlying high-dimensional distribution.

These principles have been used in work on detecting causality from interventional distributions and changing contexts [CY99, TP01a]. We will discuss the formalisms of these methods in section 2.5. Further, if we observe no change in either marginal distribution when performing an intervention on the other variable, we are forced to conclude that there is a latent (hidden) variable that must be the common cause of both [VP91].

At times, as illustrated with the absurd interventions proposed in the previous paragraph, it may not be practical, ethical or even possible to perform certain interventions. Sometimes only 'soft' interventions can be performed (ban high emission vehicles) or interventions are performed by some external agent unrelated to our causal study but we are aware of it (companies offer workers an extra paid holiday per week). Typically, one encounters a combination of such interventions when observing a system that is complex and dynamic. We believe the secret to causal learning is to be able to exploit all aspects of the dynamism in a system, where dynamism means that interventions of diff rent types take place in several contexts. Whether we perform interventions or become aware that one has been effected, a rigorous analysis of observations under these conditions can help us discover true causal structure. We provide a brief motivational example in the next section (1.4).

1.4 Motivational example: An Aircraft Monitor

Let us consider causal learning in the context of an agent monitoring a dynamic system under several contexts or external interventions. We examine a dynamic system instead of a static one for the reason that dynamic systems enable us to either conduct experiments or make observations that are akin to those collected from experiments.

Suppose that we wish to monitor an aircraft's component systems and their behavior under several conditions. As is the case with sophisticated machinery, the capabilities of laboratory design in terms of being able to provide a sufficient causal understanding of the system are limited. Modern machinery like military aircraft are constructed out of several thousand individual components each often a combination of mechanical, electrical, electronic, pneumatic and hydraulic parts. One expects to have a sufficient model of most of the aircraft and its components' expected behavior in typical and atypical conditions, based on human expert knowledge, past experience and meticulous laboratory experiments. However, this combined knowledge can still be incomplete and several aspects of the aircraft's behavior are left to be determined at the testing stage. Often, test pilots are asked to put the aircraft through a series of maneuvers that take the aircraft through several modes of operation and the results of these tests are used to establish a better understanding of the aircraft's behavior in practical operation.

Typically this is not only for the purposes of ground engineers and designers to ensure that each component works satisfactorily and as expected but also to learn some of the previously unknown or unexpected relationships between components of the aircraft system. For example, the designers may have expected that the new wing design affects the behavior of the ailerons at certain flight speeds and adjusted for it, but did not expect that the rate of fuel supply becomes intermittent as well, for the same reason. If they are unaware of the cause of this discrepancy (say, climbing angles exceeding a certain value) and hypothesize that it could be one or many of several potential causes, verification of this

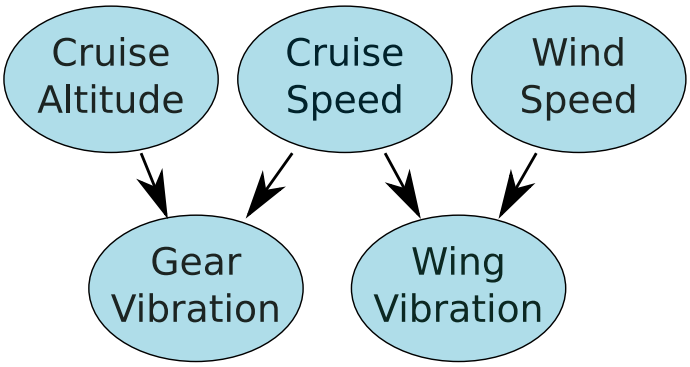


Figure 1.2: An example causal diagram between quantities monitored in an aircraft.

cause-effect relationship will involve meticulous lab tests, new flight plans, and several more man hours spent on trial-and-error.

In this situation, an automatic real-time causal learning agent mounted on the aircraft, can be very valuable. A causal learning agent that is expected to learn relationships between quantities, can record critical flight data, by segmenting it into several sections, each corresponding to a particular condition or context of aircraft operation and environment. With a causal model, we expect it to be able to answer queries about "counterfactual" contexts, and make assessments about situations yet unencountered. Additionally, we expect that it augment the test pilot's flight plan by suggesting that he take the aircraft into a new state. For example, it could request "Can you execute a 45 degree descent at Mach 2 for 30 seconds before flattening out at 10000 ft and reducing speed to 700 mph? That would help estimate effect of supersonic speed and turbulence on wing drag."

This dissertation, I address the problem of causal learning from three distinct directions. First, I improve upon existing methods for learning the equivalence classes of Bayesian networks representing a distribution from observational data alone. Second, I iintroduce a novel method of causal inference that can be used when a certain type of experimental observations are known. This method searches upstream from the experimental variable and infers the set of its parents resulting in a partition among its adjacent nodes

as parents vs. children. Thirdly, I present a new fast method to determine an experimental order among the variables that is aimed at full causal recovery with a small number of experiments. I present empirical results obtained by sampling the space of networks varying in their size and their density. I also study experimental parameters that affect performance in causality detection, including sample and network size, confidence levels for conditional independence tests and strengths of causal links. Finally, I provide hints of how to incorporate all our methods into an incremental framework that learns causal structure from streams of multivariate data.

In Chapter 2, we discuss previous work from philosophical aspects to practical issues in the field and lay down the foundations. In Chapter 3 we discuss the details and formalisms of the proposed algorithms. We run experiments on these algorithms and present our findings on several test cases in Chapter 4. Finally, in Chapter 5 we discuss how our work paves the way for future research and can be incorporated into related fields like learning dynamic causal Bayesian networks, including cyclic causal paths and extensions into continuous domains and conclude.

Chapter 2

Background

> *"You jest about what you suppose to be a triviality, in asking whether the hen came first from an egg or the egg from a hen, but the point should be regarded as one of importance, one worthy of discussion, and careful discussion at that."* - Macrobius [395-423 AD]

The goal of this chapter is to introduce the relevant background, literature and terminologies that will be used in the rest of this dissertation. This chapter is divided into six sections. Section 2.1 addresses the history and current thought on philosophical debates concerning causality. Section 2.2 delves into the relevant background on Bayesian networks and their application as causal models. Section 2.3 discusses the two main types of structure learning methods that are currently popular research topics among researchers, score based and constraint based search methods. In Section 2.4 I discuss the IC (Inductive Causation) algorithm that provides the framework for constraint based structure learning methods, its assumptions and the limits of its performance. Next, Section 2.5 introduces prior work on two main types of interventions that are relevant to this dissertation and discusses methods for causality determination that are based on them. Finally, Section 2.6 discusses some previous results on the theoretical number of experiments required for full

causal discovery under different types of interventions.

2.1 Philosophical Primer

In western philosophy, the earliest known writings on causality are by Aristotle, who identifies the Four Causes: Material, Formal, Efficient and Final [Aria, Arib]. Of the four, the type of cause that is closest to our modern notion of cause and effect is "the efficient cause", which he defines as the thing that brings something about or the primary source of the change. In India, the *Nyaya* school was an epistemology and methodology of thought that also developed some early views on a theory of causation [GauCE]. Their notion of the *Nimitta'* cause is close to Aristotle's efficient cause. Moreover, they identify conditions for causality including antecedence, invariability and unconditionality. They also identify five types of accidental antecedents which should not be confused with causal antecedents. An interesting accidental antecedent they identify is that "the co-effects of a cause are not the cause", which leads us to believe that they appreciated the difficulty in attributing orientation to a covariance and the presence of common and possibly hidden causes.

David Hume was an eminent 18^{th} century philosopher whose view was that while one can empirically verify constantly conjoined and successive events, the complete idea of causation requires a *necessary connexion* between the events that should be taken into consideration. He further argued that we can have no perceptual access to the necessary connection but we are compelled to believe in one [Hum40].

In the 20^{th} century, Max Born, a German physicist and mathematician described three assumptions that were dominant in the definition of causality, as cited in [Sow00].

1. "Causality postulates that there are laws by which the occurrence of an entity B of a certain class depends on the occurrence of an entity A of another class, where the

word entity means any physical object, phenomenon, situation, or event. A is called the cause, B the effect."

2. "Antecedence postulates that the cause must be prior to, or at least simultaneous with, the effect."

3. "Contiguity postulates that cause and effect must be in spatial contact or connected by a chain of intermediate things in contact."

In more modern literature, the best known and most thoroughly elaborated counterfactual theory of causation was proposed by David Lewis in 1973 [Lew73], which he later refined and extended [Lew86]. In this work, he describes counterfactuals and counterpart worlds. According to Lewis, a counterfactual conditional of the form *'Had I made that shot our team would have won the game.'* could be true in a world, as concrete as ours and significantly similar to it. Except that, my counterpart makes the shot rather than misses it and the counterpart of our team wins the game. Had there been a world even more similar to ours in which my counterpart makes the shot but the counterpart of our team still loses, then the counterfactual would have been false. When we speak of counterfactual possibilities we speak of what is the case in some possible world or worlds. "Actual", according to Lewis, is merely an indexical label we give to a world when we locate ourselves in it. Things are necessarily true when they are true in all possible worlds. Causation is true when its counterfactual is true in all other possible worlds. *Missing the shot* was *the actual cause* for our team to lose the game when in all other possible worlds where the shot was made, the game was won by our counterpart team.

Paul Holland identifies the *Fundamental Problem of Causal Inference* which states that it is impossible to measure the effect of two different *exposures* on the same unit. For example, if administering a medical treatment (or not) is an exposure, it is impossible to measure their isolated effects on the same patient [Hol86]. He summarizes how the different fields of economics, sociology, medicine, and philosophy deal with causal inference.

He was also the proponent of the paradigm "No causation without manipulation", which means that without experimenting upon a phenomenon systematically, one cannot truly discriminate causation from association.

This gives rise to the idea of "interventions" which is currently a popular topic of causal inference research. In [Pea00], Judea Pearl introduces a *Calculus of Interventions* and the $do(.)$ operator, which enables the empirical causal researcher to look at models under the influence of actions that force events or variables to specific values. The idea allows the statistical researcher to perform "experiments" on the data, and to reason about potential counterfactuals, under some standard assumptions. Also, Pearl argued that, using a normative assumption of Occam's Razor (principle of parsimony) some causation can be inferred without manipulation. He proposed the algorithms **IC** and **IC*** that construct and orient a causal Bayesian Network [Pea00] under these assumptions. Most interesting current research in causality follows this tradition and uses Bayesian networks as diagrams representing causality. In the next section we describe Bayesian networks in detail.

2.2 Bayesian Networks

A Bayesian network is a directed acyclic graph (DAG) consisting of nodes that represent random variables. Although it is customary to view a DAG as composed of its explicit links, it is also conceptually useful to think of a Bayesian network as a structure that encodes a conditional independence relation between pairs of variables by the *absence* of a direct link between them. A causal interpretation of the graphical structure where each directed edge represents a direct cause, leads to the notion of *causal Bayesian networks*. We shall also refer to causal Bayesian networks as *causal Models* when referring to their structure and parameters together. We shall call them *causal diagrams* or *causal graphs*, when referring to their structures alone.

Associated with its DAG structure, a Bayesian network encodes a joint probability

distribution among its component variables as a product factorization of conditional probability distributions (one per variable) along a particular ordering of these variables. Without a causal interpretation, the ordering can be arbitrary: by a recursive application of the definition for conditional probability (Equation 2.1), also known as the Chain Rule of Probability (Equation 2.2), any ordering among the variables can decompose the joint distribution [BP63].

$$P(X,Y) = P(X)P(Y|X) \tag{2.1}$$

$$\begin{aligned} P(X_1,\ldots,X_n) &= P(X_1)P(X_2|X_1)\ldots P(X_n|X_1,\ldots,X_{n-1}) \\ &= \prod_{i=1}^{n} P(X_i|X_1,\ldots,X_{i-1}) \end{aligned} \tag{2.2}$$

For example, consider the joint probability distribution among three variables A, B and C, where there is no pairwise independence among the variables (all variables are connected to each other by an arc). Using Eq. 2.1 along each ordering of the variables, the joint distribution can be factorized in six different ways, corresponding to six different Bayesian networks as shown in Figure 2.1.

1. $P(A,B,C) = P(A)P(B|A)P(C|A,B)$
2. $P(A,C,B) = P(A)P(C|A)P(B|A,C)$
3. $P(B,A,C) = P(B)P(A|B)P(C|A,B)$
4. $P(B,C,A) = P(B)P(C|B)P(A|B,C)$
5. $P(C,A,B) = P(C)P(A|C)P(B|A,C)$
6. $P(C,B,A) = P(C)P(B|C)P(A|B,C)$

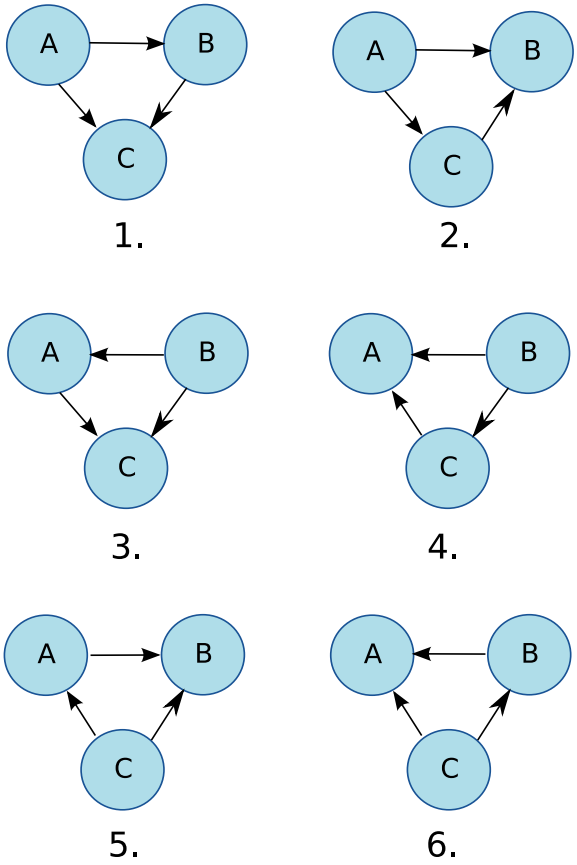


Figure 2.1: The six different configurations of a Bayesian network with 3 nodes and no conditional independence.

Now let us incorporate into this expression information about conditional independence that we can read off the DAG. Conditional independence is defined as follows:

Definition 2.2.1 (Conditional Independence)**.** *Let $X = X_1, X_2, \ldots$ be a finite set of variables. Let $P(.)$ be a joint probability function over the variables in X, and let A, B, C stand for any three subsets of variables in$X$. The sets A and B are said to be conditionally independent given C if*

$$P(A|B,C) = P(A|C) \qquad \textit{whenever} \quad P(B,C) > 0. \tag{2.3}$$

We use the notation $(A \perp\!\!\!\perp B|C)_P$ to denote the conditional independence of A and B given C, or simply $(A \perp\!\!\!\perp B|C)$ when P, the specific distribution that is referred to is unambiguous. Note that conditional independence is symmetric, i.e., $(A \perp\!\!\!\perp B|C) \implies (B \perp\!\!\!\perp A|C)$ [1]. Unconditional independence shall be denoted $(A \perp\!\!\!\perp B|\emptyset)$ or simply $(A \perp\!\!\!\perp B)$.

Applying this definition to the recursive factorization in Equation 2.2, we are left with a factorization of the joint distribution into conditional probability terms containing only a select subset of predecessors for each variable X_i, denoted PA_i:

$$P(x) = \prod_{i=1}^{n} P(x_i|pa_i). \tag{2.4}$$

We use the notational shorthand x_i to represent $X_i = x_i$, i.e., the case when the variable X_i takes the value x_i. As a general rule, throughout this dissertation, unless otherwise specified, we use uppercase letters without subscripts (e.g. X) to denote sets of variables, uppercase letters with subscripts (e.g. X_i) to denote singleton variables and lower case letters (x or x_j) to denote the values these variables (or sets) can take.

The set PA_i is called the 'parents' or Markovian Parents of node X_i and is defined below. This becomes easier to visualize when we view each variable as a node in a DAG, and that each of the Markovian parents of that variable give rise to arcs directed towards the variable, thus describing the structure of the Bayesian network.

Definition 2.2.2 (Markovian Parents)**.** *Let $X = X_1, \ldots, X_n$ be an ordered set of variables, and let $P(v)$ be the joint probability distribution on these variables. A set of variables PA_j is said to be* Markovian parents *of X_j, if PA_j is a minimal set of predecessors of X_j that*

[1] There are other properties of conditional independence apart from symmetry, but they are not trivial or very intuitive and we will not refer to them here. A partial list of properties of conditional independence relations discovered so far have been summarized by Spohn et al. in [SPB94] and also by Pearl and Geiger in [GP93]. Despite several promising advances and contributions by several researchers during the past three decades, a completeness result still eludes.

renders X_j independent of all its other predecessors. In other words, PA_j is any subset of $X_1, \ldots, X_{j-1}$ satisfying

$$P(x_j|pa_j) = P(x_j|x_i, \ldots, x_{j-1}), \tag{2.5}$$

such that no proper subset of PA_j satisfies Eq. 2.5.

Once, the Bayesian network is described this way, the order of the variables becomes irrelevant. When we are given P and G, we can test whether P decomposes into the product as described by G. Another advantage of Bayesian networks is that they provide a tractable representation of the joint distribution.

Consider that the joint distribution we need to represent is for a set of n binary variables. The distribution would be a real numbered probability value for all possible configurations of these variables, demanding a storage of $O(2^n)$ floating point memory locations. Except for very small n, this is intractable. However, consider the same distribution represented by a Bayesian network. From the factorization mentioned in Eq. 2.4, we know we need n conditional probability tables. If the maximum in-degree of the DAG is k, the total space required is $O(n2^k)$ (the storage for the DAG needs only $O(nk)$ and can be ignored). If k is reasonably small and invariant to n, as we expect for most real applications, then there is an enormous space savings achieved.

Definition 2.2.3 (Markov Compatibility)**.** *If a probability function P admits the factorization of Eq. 2.4 relative to DAG, G, we say that G represents P, that P and G are compatible, or that P is Markov relative to G.*

Compatibility between DAGs and probability functions is the key to statistical modeling and is a necessary and sufficient condition for a DAG G to explain empirical evidence represented by P. If each conditional probability satisfies a set of conditional independence relationships, Markov compatibility ensures that these can be read off the DAG by a criterion known as *d*-separation (*d* stands for *directional*).

Definition 2.2.4 (*d*-Separation)**.** *A path p is said to be* d-separated *(or* blocked*) by a set of nodes Z if and only if*

1. *p contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle node m is in Z, or*
2. *p contains an inverted fork (also known as a "collide") $i \rightarrow m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.*

A set Z is said to d-separate X from Y if and only if Z blocks every path from a node in X to a node in Y.

We will use the notation $(X \perp\!\!\!\perp Y|Z)_G$ to denote that X and Y are d-separated by Z in DAG G

The correspondence between d-separation and its probabilistic analogue is summarized by this theorem from [Pea88].

Theorem 2.2.1 (Probabilistic Implications of *d*-separation)**.** *If X and Y are two sets of variables* d*-separated by set Z in a DAG G, then X is independent of Y conditional on Z in every distribution compatible with G. Conversely, if X and Y are not d-separated by Z in a DAG G, then X and Y are dependent conditional on Z in at least one distribution compatible with G.*

Let us revisit the example of unconstrained Bayesian networks on the variables A, B and C shown in Fig.2.1. Using these results, we can now apply the conditional independence statement $(A \perp\!\!\!\perp B|C)_P$ for all P compatible with G, leading to the implication that $(A \perp\!\!\!\perp B|C)_G$. Using Definition2.2.4, for d-separation, we get the set of DAGs in Fig.2.2, each compatible with the conditional independence statement $(A \perp\!\!\!\perp B|C)$. The introduction of the conditional independence statement reduced the number of Bayesian networks that encode the distribution from 6 (in Fig.2.1) to 3 (in Fig.2.2).

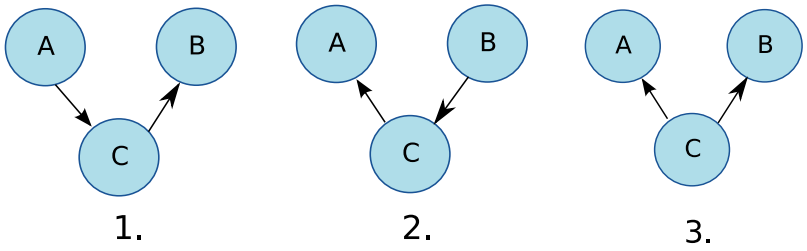


Figure 2.2: The configurations of a Bayesian network with 3 nodes with $A \perp\!\!\!\perp B|C$

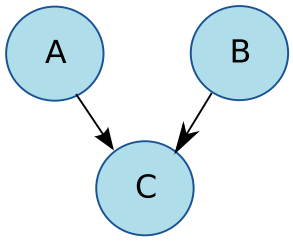


Figure 2.3: The Bayesian network with 3 nodes and $A \perp\!\!\!\perp B$

Now, consider that we introduce the unconditional independence statement $(A \perp\!\!\!\perp B|\emptyset)$. The number of Bayesian networks that encode this reduces to a single unique network (Fig. 2.3).

The key insight that we learn from this is that when a conditional independence statement $(A \perp\!\!\!\perp B|S_{AB})$ is introduced, the link between A and B disappears in all cases, but the nature of the separating set S_{AB} further determines the set of Bayesian networks that are compatible with that statement. From this example, we see that common neighbors of A and B that are not in S_{AB} form *v-structures* or colliders with A and B, i.e. arcs from A and B converge on all $C \notin S_{AB}$ where $(C \not\perp\!\!\!\perp A|S_{AC}) \wedge (C \not\perp\!\!\!\perp B|S_{BC})$ for any S_{AC} and S_{BC}.

This provides us the intuition for the notion of observational equivalence, as specified by the following theorem from [VP90].

Theorem 2.2.2 (Observational Equivalence)**.** *Two DAGs are observationally equivalent*

if and only if they have the same skeletons and the same set of v-structures, that is, two converging arrows whose tails are not connected by an arrow.

From this theorem we see that a certain equivalence class of Bayesian networks are observationally indistinguishable. All networks belonging to an equivalence class have the same undirected skeleton, and the same "unshielded colliders" and there are no statistical tests (however perfect) that can be performed on distributions (or from data generated from these distributions) that can ascertain which one of these networks in the equivalence class represents the true causal relationships. However, if experimental data is available, then one can further reduce the size of the equivalence class. With a sufficient number of experiments we can narrow down the set of possible networks to a single causal network [ES06, ES07]. In the next section we present current methods in structure learning. We then have further discussion on this topic Section 2.5 onwards.

2.3 Structure Learning in Bayesian Networks

Bayesian network structure learning from data is hard particularly because of the extremely large search space. The number of Bayesian network structures (DAGs) over n nodes is given by Robinson's formula [Rob76]:

$$G(n) = \begin{cases} 1 & \text{if n = 0} \\ \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} G(n-i) & \text{otherwise.} \end{cases} \tag{2.6}$$

Table 2.3 shows the value of $G(n)$ for the first 10 values of n. Fig. 2.4 shows the growth of the logarithm of the number of DAGs with respect to number of nodes. This formula tells us that the space of Bayesian networks is super-exponential w.r.t. the number

n	$G(n)$
1	1
2	3
3	25
4	543
5	29281
6	3781503
7	1.1388×10^9
8	7.8370×10^{11}
9	1.2134×10^{15}
10	4.1751×10^{18}

Table 2.1: The number of DAGs $G(n)$ as a function of number of nodes, n.

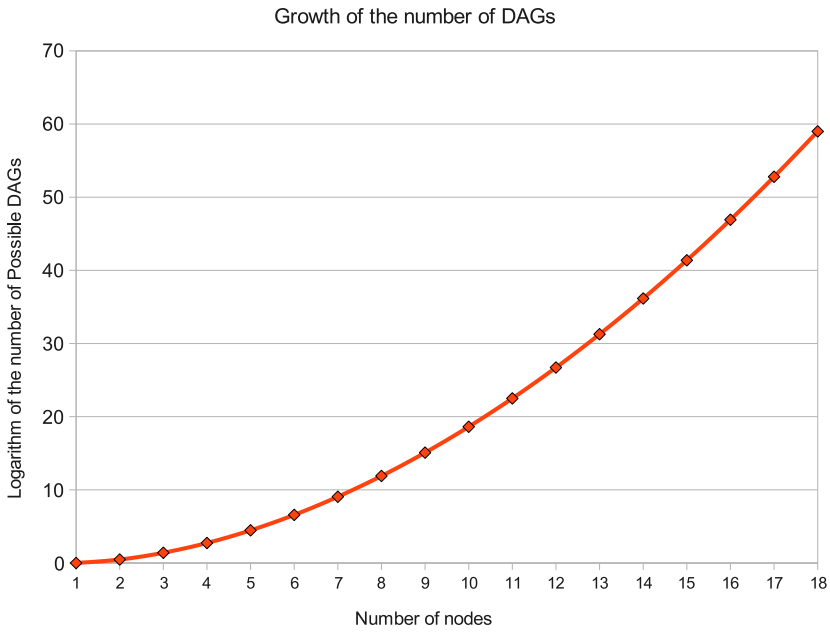


Figure 2.4: The greater than linear growth of the logarithm of the number of DAGs calculated by Robinson's formula.

of variables. Any kind of strategy to exhaustively search the space of networks for any realistic problem becomes intractable.

There have been two main approaches to Bayesian network structure learning.

1. Score based search

2. Constraint based search

Score based search methods mostly follow a common format. A score is a measure of quality of fit between a given Bayesian network and data. Typically, it is an indicator of the likelihood that the observations were generated by the Bayesian network. This measure of quality on Bayesian network structure, allows us to discriminate between the individual networks in the search space with respect to the applicability of each network in a particular domain. Usually, we desire that the network structure faithfully captures the underlying dependencies in the data and is a good explanation for the data. In other words, we look for a "goodness of fit" score which maps a value to each network in the search space, and this is exploited by search algorithms to find maxima on the landscape of scores over the search space. There are many types of scores used in the literature, but the most common and successful ones are based on the Minimum Description Length (MDL) principle and the Bayesian Information Criterion (BIC). The graph space is described in terms of neighborhoods of edge additions, removals and reversals. The search starts at some random or heuristically chosen graph, and executes a greedy search in this neighborhood until no further improvement in score is obtained. To avoid getting trapped in local minima, several techniques are applied, including random restarts, TABU search, simulated, annealing and data bootstrapping.

The MDL principle [Ris78] is a formalization of *Occam's Razor* also known as the "Principle of Parsimony". It is based on the idea that the best model of a database is the model that minimizes the sum of the length of the encoding, or in this case, the Bayesian network.

Some of the earliest work on learning structure for knowledge representations was done in 1968. Chow and Liu [CL68] proposed the first ever algorithm that learns a tree structure that maximally approximates the database distribution. Their method, based

on a minimum difference in information, finds the maximum likelihood estimate of the distribution when the structure is restricted to trees.

More recently, Buntine proposed an algorithm that searches the space of all DAGs, using a greedy blind search [Bun91]. An important point about this work is that it had an incremental flavor, assumes an existing knowledge base and the algorithm augments it with more rules when provided with more data. Cooper and Herskovitz [CH92] took the Bayesian approach and proposed an algorithm that when given a variable ordering delivers a DAG. An interesting feature of their method is that they restrict the search to finding one single network among an equivalence class of DAGs. Later, Castelo and Kocka [KC01] emphasize that the *Model inclusion* principle imposes an ordering among DAGs and they propose an improvement to Bayesian network learning that avoids the problem of local minima.

In 2004, Koivisto and Sood presented an exact algorithm for Bayesian structure discovery [KS04]. While being exact, it was the first algorithm with less than super-exponential complexity with respect to the number of nodes. They also assert that in some special cases where suitable restrictions can be placed on the structures, exact methods can be made feasible or can be combined with inexact methods to create a trade-off between exactness and feasibility.

Batch learning methods have their limitations with respect to database size and complexity. Another consideration is that in realistic settings learning algorithms have to be designed to operate incrementally, using "online" learning techniques. These algorithms operate on the premise that the learning task uses only finite memory and finite computational resources, and therefore can neither store arbitrarily large amounts of data nor can it relearn from scratch each time it updates its knowledge. Typically, the aim of most incremental algorithms is to visit each record just once.

Buntine's batch algorithm [Bun91] has an incremental flavor. Buntine also provides

guidelines for an incremental version. Lam and Bacchus propose a technique [LB94] revising Bayesian networks incrementally based on improving the description length (DL) of a subgraph, and also show how this reduces the DL of the whole graph if no cycles are introduced. Friedman and Goldszmidt [FG96] propose and analyze three approaches: the first is to store all the data and simply relearn everything from scratch; the second approach uses a single structure for storing prior data; the third approach is a combination of the two and exhibits natural trade-offs.

In the related domain of undirected graphical models, Domingos and Kok propose both a batch and revision based algorithm for Markov Logic Networks based on relational databases [KD05]. Markov logic networks (MLNs) combine logic and probability by attaching weights to first-order clauses, and view these as templates for features of Markov networks. Combining ideas from inductive logic programming (ILP) and feature induction in Markov networks, their algorithm performs a beam search of the space of clauses, guided by a weighted pseudo-likelihood measure.

In [Alc05], Alcobé proposes two general search heuristics that convert batch learning algorithms to incremental ones. One of their heuristics, the Traversal Operators in Correct Order (TOCO) ensures that the structure will be revised only if it is invalidated by new data; when it must be revised, the learning algorithm does not begin from scratch. The second Reduced Search Space (RSS) heuristic, uses the knowledge gathered from previous learning steps stating that structures that had very low quality in past learning steps will still have low quality with respect to the new dataset in the current learning step.

While all these methods enjoy varying degrees of success in finding high scoring graphs, the Bayesian methods are compatible with the idea of stability. Methods of incorporating mixtures of observational and experimental data to find pairwise causal links have also been investigated under the Bayesian approach [CY99]. They tend to operate well for small datasets but suffer when there are hidden variables and large networks. Evaluating the Bayesian score involves computing an expensive integral (summation) in

discrete domains. As a consequence of Observational Equivalence (Theorem 2.2.2) we can expect to find a graph that belongs to the same equivalence class as the causal network, if not the true causal network itself. This is not guaranteed by all of the score-and-search methods and graphs.

Constraint Based search methods, are an interesting alternative to greedy search techniques. These methods start with an unconstrained structure (complete and undirected graph) and apply successive constraints on this structure as implied by Conditional Independence Statements (CISs) to arrive at an intermediate undirected skeleton, and then orient the edges as implied by the CISs (d-separation implications) and DAG acyclicity. The earliest algorithm in this family of methods is the IC algorithm (Inductive Causation) described by Verma and Pearl [VP91], and it guarantees that the partially directed acyclic graph (PDAG) describes the equivalence class of structures that represent the given CIs. In this dissertation we take the constraint based approach and discuss the IC algorithm in the following section.

2.4 The IC algorithm

The IC algorithm relies on three assumptions: the Causal Markov Condition, Stability, and Sufficiency. We briefly discuss these assumptions.

Theorem 2.4.1 (The Causal Markov Condition [VP91])**.** *Every Markovian causal model M induces a distribution $P(x_1, \ldots, x_n)$ that satisfies the parental Markov condition relative to the causal diagram G associated with M; that is, each variable X_i is independent of all its non-descendants, given its parents PA_i in G.*

Intuitively, ignoring a variable's effects, all relevant probabilistic information about a variable that can be obtained from a system is available from its causes. This is similar to an interpretation of a first-order Markov process; knowledge about the current state allows

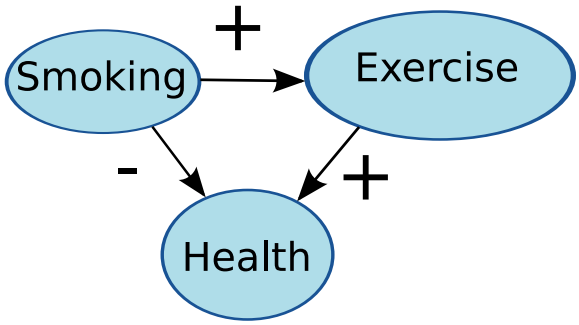


Figure 2.5: An example of an *unfaithful* causal graph. The +/- links correspond to positive vs. negative influences.

us to predict its next states, but it is not necessary to know how the process came to be in the current state. While the Markov condition (Definition 2.2.3) allows us to use an elegant graphical theory to interpret a probability distribution, the causal Markov condition goes a step further and interprets the DAGs causally. It makes the central assumption that the Markov condition and *d*-separation are in fact the correct link between causal structure and probabilistic independence.

Definition 2.4.1 (Stability)**.** *Let $I(P)$ denote the set of all conditional independence relationships embodied in P. A causal model $M = \langle D, \Theta_D \rangle$ generates a stable distribution if and only if $P(\langle D, \Theta_D \rangle)$ contains no extraneous independencies, that is, if and only if $I(P(\langle D, \Theta_D \rangle)) \subseteq I(P(\langle D, \Theta'_D \rangle))$ for any set of parameters Θ'_D*

Although Pearl defined the concept of *stability* as an interpretation of Occam's razor w.r.t. the causal Markov condition [Pea00], it is more commonly referred to as the Faithfulness criterion, an equivalent definition introduced by Clark and Glymour[SGS00, SGS01].

Definition 2.4.2 (Faithfulness)**.** *Let G be a causal graph and P a probability distribution generated by G. $\langle G, P \rangle$ satisfies the Faithfulness condition if and only if every conditional independence relation true in P is entailed by the Causal Markov condition applied to G.*

Intuitively, while the causal Markov condition ensures that any distribution P produced by the graph G has the corresponding probabilistic independencies implied by applying

d-separation to G., faithfulness ensures that P has exactly those and no additional independence relationships. Cartwright [Car83] introduced the following example of a causal graph that is *unfaithful*. Consider a graph of three variables $Smoking$, $Exercise$ and $Health$ as shown in Figure 2.5. Let us assume $Health$ is positively affected by $Exercise$ and negatively by $Smoking$, as is generally believed. Further, assume that $Smoking$ has a positive effect on $Exercise$ (absurd as this may be). If the parameters of these relationships are just "right", such that the positive effect of $Smoking$ (through $Exercise$) cancels its direct negative effect, $Smoking$ and $Health$ might become probabilistically independent. While it is acknowledged that such graphs do violate the Faithfulness assumption, it is also believed that such graphs are extremely rare in practice and that such contrived parametrizations (where Nature is acting like a cruel adversary) are 'unstable' and that they do not prevail across multiple instances. Faithfulness allows us to ignore all these cases in causal analysis and is widely accepted as a reasonable assumption.

Definition 2.4.3 (Causal Sufficiency)**.** *The set of measured variables V include all of the common causes, if any, of each pair of variables in V.*

The final assumption is causal Sufficiency, which makes an assumption on the ability to make all relevant *measurements*, i.e., all the common causes of all measured variables. In other words, Causal Sufficiency assumes there are no latent, hidden variables that could be the cause of more than one variable measured in the system. This is perhaps the most unrealistic assumption among those presented, as it is easy to imagine several practical situations when observations are *insufficient*. However, there is a large class of problems that satisfy this requirement and it is worth pursuing this approach. Moreover, algorithms that make the Sufficiency assumption are conceptually simpler and provide the framework for extension to more sophisticated algorithms that can deal with hidden variables. For example the IC*, algorithm, extends IC to the case where the Sufficiency assumption is not made. We will however assume Sufficiency throughout this dissertation, and will not discuss further the implications of not assuming Sufficiency [SMR95, SG09].

The following algorithm takes as input a stable probability distribution P generated by some underlying DAG D_0 and outputs a PDAG that represents the equivalence class of D_0 [VP91].

Algorithm *IC (Inductive Causation)*
Input: P, a stable distribution on a set of V variables
Output: $H(P)$, a PDAG

1. For each pair of variables a and b in V, search for a set S_{ab} such that $(a \perp\!\!\!\perp b | S_{ab})$ holds in P. Construct an undirected graph G such that vertices a and b are connected with an edge if and only if no set S_{ab} can be found.
2. For each pair of nonadjacent variables a and b with a common neighbor c, check if $c \in S_{ab}$. If it is, continue, otherwise add arrowheads pointing at c. (i.e., $a \rightarrow c \leftarrow b$).
3. In the partially directed graph that results, orient as many of the undirected edges as possible subject to two conditions: (i) the orientation should not create a new *v*-structure; and (ii) the orientation should not create a directed cycle.

It has been shown [Mee95, Zha08] that there exist a set of rules of orientation (for step 3) which upon repeated application, guarantee that *all* arrows that are common to the equivalence class of D_0 will eventually be oriented. The IC algorithm, therefore is a very powerful and valuable tool in causal structure learning, and provides a guarantee of the quality or closeness of the structure found to the actual causal structure. In a typical causal learning problem the IC algorithm orients a large fraction of the edges and allows us to focus on experiments that focus on learning only those edges that are still undirected. Interventions, experimental data and dynamism, are some of the concepts that can be used to learn further causal information about the system. In section 2.5 we characterize interventions, discuss some of the existing methods for causal learning from interventions, and finally end the chapter with a section on the number of interventions required for full causal learning.

2.5 Interventions

Pearl introduced the *calculus of interventions* as a theory of causality by manipulability [Pea95, Pea00]. In other words, the test for a causal connection is by the ability to force a change on the effect by manipulation of its causes. Manipulating a variable or a set of variables is called an intervention, and is represented by the $do(.)$ operator. For example, $do(X = x)$ represents the *atomic intervention* of setting a single variable X to the value x. A distribution measured on another variable Y under this intervention is represented as $P(Y|do(X = x))$. Alternatively, throughout this dissertation, we use the notational shorthands $P(Y|\hat{x})$ or $P_x(Y)$ to represent atomic interventions.

In contrast to the conditional operator in probability theory, the atomic intervention $P(Y|\hat{x})$ does not represent the distribution on Y when X is given to be x, but rather it represents the distribution on Y when X is "set" to x. The term $P(Y|X)$ represents the distribution on Y given that we observe X where X varies freely according to its governing distribution. Whereas the "causal effect" term $P(Y|do(X))$, represents the distributions on Y when X is held at fixed values. It has the effect of neutralizing the effect of X's predecessors on X.

Definition 2.5.1 (Causal Effect)**.** *Given two disjoint sets of variables, X and Y, the causal effect of X on Y, denoted either as $P(y|\hat{x})$. or as $P(y|do(x))$, is a function from X to the space of probability distributions on Y. For each realization x of X, $P(y|\hat{x})$ gives the probability of $Y = y$ induced by deleting from the model of 1.2 all equations corresponding to the variables in X and substituting $X = x$ in the remaining equations.*

One way to conceptualize $P(Y|do(X_i = x_i))$ is to consider the effect of the intervention on a causal diagram. The intervention on X_i effectively severs all the parental arrows from $pa_i)$ to X_i. Thus, the atomic intervention renders the intervened variable independent of its normal causes. Another notation used is to consider an intervention by the introduction of an interventional node F_i as a parent of X_i, (Figure 2.6) which takes two values

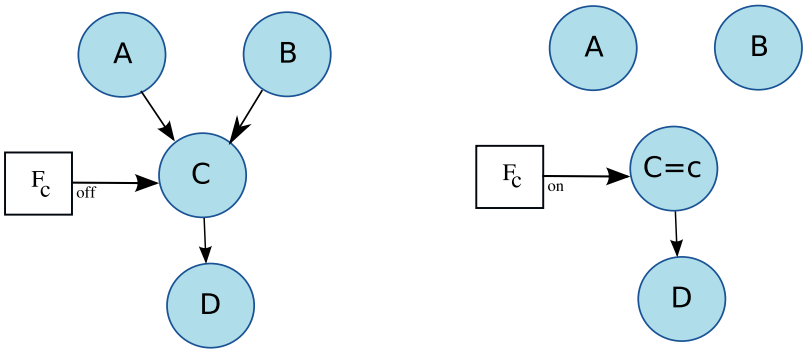


Figure 2.6: An intervention node F_c that represents an atomic intervention performed on variable C

'on' and 'off'. The 'off' state represents no intervention and the 'on' state refers to the atomic intervention $do(X_i = x_i)$.

Recall the factorization of the joint probability represented by the Bayesian network (Equation 2.4) in terms of its conditionals. When an atomic intervention $do(x_i')$ is performed on a variable X_i setting them to the set of values x_i', the conditional that are not consistent with the interventions disappear, and when they are consistent they become unity. Let the interventional node be an additional parent to each of the intervened variables ($pa_i' = pa_i \cup F_i$) . The altered conditional probabilities of this augmented Bayesian network can now be written as:

$$P(x_i|pa_i') = \begin{cases} P(x_i|pa_i) & \text{if } F_i = off \\ 0 & \text{if } F_i = do(x_i') \text{ and } x_i \neq x_i' \\ 1 & \text{if } F_i = do(x_i') \text{ and } x_i = x_i' \end{cases} \tag{2.7}$$

The effect of the intervention $do(x_i'))$ is to transform the pre-interventional probability distribution $P(x_i, \ldots, x_n)$ to the the post-interventional distribution $P(x_i, \ldots, x_n|\hat{x}_i') = P'(x_i, \ldots x_n|F_i = do(x_i'))$. In terms of the conditional probability factors of the pre-intervention distribution, we can write the *truncated factorization* formula 2.8:

$$P(x_i, \ldots, x_n | \hat{x}'_i) = \begin{cases} \prod_{j \neq i} P(x_j | pa_j) & \text{if } x_i = x'_i \\ 0 & \text{if } x_i \neq x'_i \end{cases} \tag{2.8}$$

So far, we have only talked about the *atomic intervention* where we set the value of the intervened variable X_j to a fixed constant. This could be termed a **hard intervention**. We can conceive of several practical situations where such a hard intervention may not be possible, but perhaps we are able to make a subtler, more general change to X_j, a **soft intervention**. If we replace the "mechanisms" that determine X by another equation such that $PA^*(X)$ are now the new parents of X we can write the new joint distribution P^* as

$$P^*(x_1, \ldots, x_n) = P(x_1, \ldots, x_n) \frac{P^*(x_i | pa_i^*)}{P(x_i | pa_i)} \tag{2.9}$$

Note that the parents of x_i, pa_i is replaced by pa_i^* indicating that the set of parents could potentially be different. A simplifying assumption that can be applied is that pa_i does not change across the interventional boundary, but only the parameters of the conditional distribution $P(x_i|pa_i)$ on the *focal variable* x_i changes to $P^*(x_i|pa_i)$ [TP01a]. In other words, the structure of the causal model remains invariant while the parametrization is altered by the "mechanism change".

There are several other models of intervention discussed in recent literature, including *uncertain interventions, fat-hand interventions and imperfect interventions* [EM00, TKP06]. In this dissertation, we employ only hard and soft interventions. In the next section we will discuss how interventions are useful in determining causal structure.

2.5.1 The Power of Interventions

In section 2.2 we discussed Observational Equivalence (Theorem 2.2.2) and how when given a distribution that echoes a set of conditional independence statements, there are several indistinguishable networks that form a Markov equivalence class given the distribution. Under interventions, however, we are able to reduce the size of this equivalence class and under the right set of interventions we are able to recover the true causal structure. For example, consider the CANCER network [FMR98] as shown in Figure 2.7. Reading off the graph by the rules of *d*-separation we see that it encodes the following set of CIs : $\{(A \perp\!\!\!\perp D|B,C)$, $(A \perp\!\!\!\perp E|C)$, $(B \perp\!\!\!\perp C|A)$, $(B \perp\!\!\!\perp E|A)$, $(D \perp\!\!\!\perp E|C)\}$. Given only this, and using theorem 2.2.2, we can now draw the structures with the same skeletons and same set of *v*-structures as shown in Figure 2.8 that are observationally (or Markov) equivalent to the CANCER network. With observational data, this equivalence class represents the limit of our causal inference.

Each intervention, however, helps us determine the orientations of the edges to the neighbors of the intervened variable [ES06, ES07, TP01a, TP01b]. Figure 2.9 shows the set of structures equivalent under an intervention on B . Structure iv from figure 2.8 has the wrong parents for B and hence gets eliminated. Some other non-neighborhood edges might also be forced towards an orientation due to *d*-separation and DAG acyclicity. The variable chosen for intervention also plays a significant role in the number of edges that get oriented. Figure 2.10 shows the unique causal structure that can be determined from an intervention on A. Interventions on A determine the arcs out of A to its neighbors, $A \rightarrow B$ and $A \rightarrow D$ and the arc $C \rightarrow E$, since we cannot introduce new *v*-structures. In general, given interventional data, we can eliminate several networks out of the observational equivalence class and determine the interventional equivalence class, as proved in [TP01a]. In chapters 3 and 4, we will discuss algorithms and empirical considerations that extend previous methods that do this.

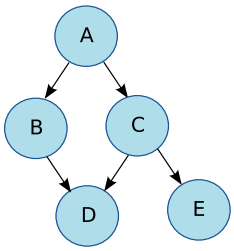


Figure 2.7: The CANCER network

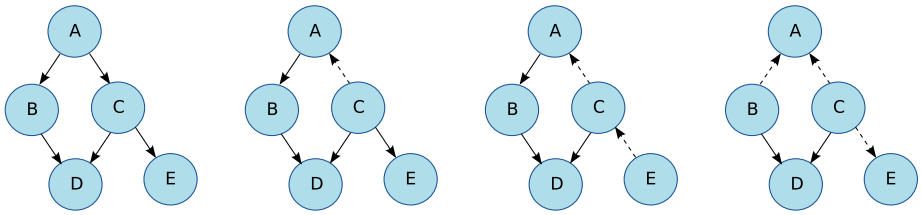


Figure 2.8: Markov equivalent structures of the CANCER network. Dashed edges are the edges that differ from the original network.

2.6 On the Number of Experiments

We saw in section 2.5.1 that different interventions determine different sets and numbers of orientations in the causal model. Some interesting questions arise naturally from that discussion.

1. How many interventions are needed to determine full causal structure?

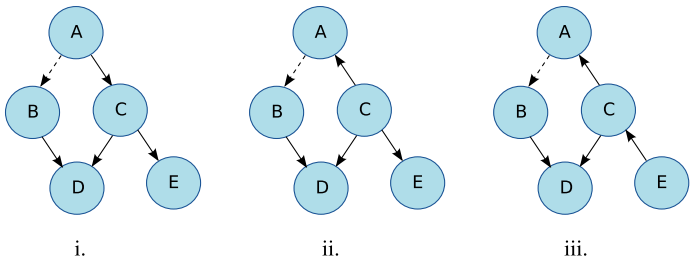


Figure 2.9: Markov equivalent structures of the CANCER network under an intervention on B. Dashed edges are a result of the intervention.

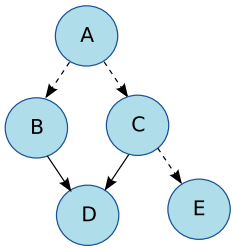


Figure 2.10: Unique causal structure of the CANCER network recovered under an intervention on A. Dashed edges are a result of the intervention.

2. Is there an optimal ordering of these interventions so that we can determine maximum causal structure soonest?

3. Does the type of intervention affect the number of interventions required?

4. If there are some specific links whose orientations we are interested in more than others, which and what kind of interventions should be prioritized?

Tian and Pearl [TP01a], consider a transition sequence (TS) of soft interventions on single focal variables at each transition to detect marginal changes in descendants. Relying on intuition, they construct this sequence as a transition and construct a Marked Order Graph (MOG) that is then used to constrain learning algorithms. In a series of papers, Eberhardt, Scheines and Glymour address these questions [EGS05, ES06, ES07, Ebe06, Ebe08, EGS06, Ebe10]. In [EGS05, EGS06], they show that under the usual assumptions of Faithfulness, Markov compatibility, causal Sufficiency, and perfect data, $N-1$ experiments suffice and in the worst case are necessary to determine the causal links among $N > 2$ variables. An experiment here corresponds to the equivalent of randomized controlled trial (a perfect intervention) on one variable at a time. But this bound does not hold when $N > 4$ and experiments are performed on more than one variable at a time. In fact, surprisingly, this bound reduces, to $\lfloor log_2(N) + 1 \rfloor$ when multiple simultaneous experiments are allowed [EGS05]. Further, parametric interventions that do not alter the structure of the model are more powerful under correlational tests of causal pathways,

and the number of experiments required reduces to one, where the parameters of all the variables are simultaneously changed.

Eberhardt [Ebe08] provides results of a simulation to support a conjecture that the worst case number of experiments necessary and sufficient to discover a causal graph uniquely given its observational Markov equivalence class as a function of the largest clique in the Markov equivalence class. An interesting extension to this approach addresses the problem with a game theoretic focus [Ebe10]. The problem of causal discovery is framed as a game of the Scientist against Nature, in which Nature attempts to hide its secrets for as long as possible, and the Scientist makes her best effort at discovery while minimizing the cost involved in running experiments. A key limitation of these results is that they rely on the assumption that it is possible to perform these experiments in the first place. Additionally, some experiments may be more difficult than others, or be very expensive, and some may be unethical or impractical to perform. Nevertheless, they provide valuable guidance to a causal learner in making choices while determining an order among the possible set of interventions.

In chapter 3, based on the background of chapters 1 and 2, I present the primary hypotheses of this dissertation and how they will be supported.

Chapter 3

Three Improvements to Algorithms for Causal Learning

"Frustra fit per plura quod potest fieri per pauciora."

- Franciscan friar William of Ockham, 14^{th} Century

"Simplicity is the ultimate sophistication."

- Leonardo Da Vinci ,15^{th} Century

"We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances"

-Sir Isaac Newton, 17^{th} Century

"Make everything as simple as possible, but not simpler."

- Albert Einstein, 20^{th} Century

"When you have eliminated the impossible, whatever remains, however improbable, must be the truth."

- Sherlock Holmes (Sir Arthur Conan Doyle) in The Sign of the Four [Doy90].

Chapter 1 introduced the general idea of learning causal structure and its applicability across several types of domains. In Chapter 2, presented the relevant background related

to this problem and discussed formal frameworks for model inference, including causal Bayesian networks, and presented related definitions. I also introduced the IC algorithm for inferring partial causal structure from observational data and presented some types of interventions and their utility in augmenting causal structure.

In this chapter, I present the three hypotheses that make up my dissertation research.

Hypothesis 1 (Robust Constraint-Based Structure Search)**.** *Constraint-based structure search algorithms that operate on finite samples are vulnerable to faulty statistical tests that are used to infer conditional independence information. I propose a new algorithm based on a tunable parameter that provides an alternative between greedy and conservative methods of choosing independence constraints that is robust to faulty tests.*

In section 3.1 of this chapter I first introduce the PC algorithm, a greedy and computationally tractable adaptation of the IC algorithm. To put in perspective the practical means of determining conditional Independence statements (CIs), in section 3.2 I present a mutual information based empirical technique for determining CIs from multinomial data samples. I then discuss a conservative version of the PC algorithm, CPC_{or}, that relaxes the *faithfulness* assumption and accounts for certain types of problems in a set of determined CISs statements. In section 3.4, I present the *soft*-CPC algorithm ($sCPC_{or}$), a less conservative approach that trades off between CPC_{or} and PC_{or}. I present arguments that justify this trade-off in lieu of faulty CISs from sampled data. In section 4.2, I present the results of $sCPC_{or}$, comparing it to PC_{or} and CPC_{or}, for several networks across sample and network sizes.

I then move on to algorithms that exploit experimental distributions. I handle two types of experimental distributions, i.e., distributions due to perfect multi-variable interventions and distributions due to soft interventions on single variables.

Hypothesis 2 (Parent Detection)**.** *Perfect interventions on a set of variables sever the causal connections from the parents of the intervened variables to the intervened set,*

which exhibits a certain specific type of difference between the pre-interventional and post-interventional distributions. I present an algorithm that exploits this difference to determine the most likely parental connections of the interventional variables.

In section 3.5, I present the *parentalSearch* algorithm that exploits the difference between pre- and post-interventional distributions of perfect interventions on multiple variables to execute a search upstream in the causal order. This search is tractable as it is constrained among the undirected neighbors of a causal graph whose adjacency information is already determined through constraint based methods. In section 4.3, I show the performance of this algorithm across varying sample sizes, network complexity and the choice of interventional sets.

In section 3.5.1, I then present the *approximateKLD* technique that I developed that is required by the *parentalSearch*. *approximateKLD* is an algorithm that estimates the KL-divergence of two high-dimensional distributions entirely from data samples. While it suffers from not being accurate to the true KLD, I show empirically that it performs well as a relative metric (this suits *parentalSearch*), which is sometimes what algorithms need. Further it has the advantage of being computationally tractable. Section 4.4 presents the corresponding results.

Hypothesis 3 (A interactive algorithm to prioritize interventions)**.** *While soft interventions do not sever any causal connections in the causal model but instead introduce a change in the governing mechanisms. Pre- and post-interventional distributions of soft interventions can be thought of as a randomized controlled trial and they mirror the philosophy of causality through manipulability. Combining from constraint-based methods, I present a novel and fast incremental algorithm that can also be deployed interactively that learn orientations to the PDAGs from parametric interventions by minimizing the number of interventions required.*

In section 3.6 I present an interactive algorithm, *interactiveCausalDiscovery* that starts with the pre-interventional distribution, and suggests to a human experimenter, a prioritized set of interventions that she may choose to perform. Upon accepting this choice, and the corresponding data, the algorithm incorporates this information into the causal model and returns an updated set of choices to the human. This proceeds until all orientations are determined. In cases where it is possible to automate the experimentation and data collection process, the algorithm would simply choose the highest priority experiment at each iteration and proceed. I also discuss the conjecture on the worst case minimum number of experiments required and its connection to graphs of different density.

The algorithm supporting hypotheses 1 and 2 complement hypothesis 3. When the interventions are parametric, *interactiveCausalDiscovery* deploys a constraint based method (like $sCPC_{or}$) by augmenting the causal model with intervention nodes to detect orientations. When given hard interventions *parentalSearch* executes an "upstream search" in the causal order. Note that other methods like descendant detection with marginal distributions can also be used, with the *interactiveCausalDiscovery* algorithm, but I do not discuss the details in this dissertation.

Section 4.5 contains the results of applying this technique to causal graphs of varying size and complexity. Finally, I discuss how a combination of the above algorithms can be used in an incremental fashion in practical causal learning, to conclude chapter 3. The results of these combined experiments concludes 4.

3.1 The PC Algorithm

The PC algorithm, named after its creators P. Spirtes and C. Glymour [SGS01], improves on the basic idea of the IC algorithm by exploiting the sparseness of the causal graph. The IC algorithm has a subset search routine where, for a pair of vertices $\{a, b\}$, a separating set $S_{a,b} \subseteq V \setminus \{a, b\}$ is to be found. The powerset of the remaining vertices, $2^{V \setminus \{a,b\}}$, specifies

the exponential search space. An edge (a, b) exists in the causal graph only if there exists no conditional independence between the two vertices conditioned on any subset of the remaining vertices. For graphs that have a degree much smaller than than $|V|$, we can get significant reduction in the time complexity by restricting the search for S_{ab} only to nodes that are still adjacent to a and b. Indeed, the PC algorithm enjoys, polynomial complexity in graphs of finite degree, as it systematically explores the search space in sets of increasing cardinality, removing the edge (a, b) as soon as a separating set is found, automatically preferring smaller separation sets to larger ones, following the principle of parsimony.

Following [KB07], we discuss the PC algorithm as proceeding in two stages.

1. PC_{sk} : The first stage of the algorithm that determines the undirected graph (skeleton) among the variables.

2. PC_{or} : The second stage which uses a set of rules to orient as many undirected edges as possible.

For the sake of clarity in describing the algorithm, the first stage assumes perfect knowledge about the set of all conditional independence relationships among the variable set, V. In other words, we assume that we are given a set S_{CIS} that can be queried for membership of statements of the type: $(A \perp\!\!\!\perp B|C)$, where A and B are variables and C is such that $C \subseteq V\ A, B$. However, in the next subsection we will relax this assumption and describe an empirical method to determine these relationships.

Algorithm PC_{sk}

Input: V, S_{CIS}

($*$ Vertex set and set of conditional independence statements $*$)

Output: $G(V, E), S$

($*$ An undirected graph, set of separating sets $*$)

1. $G(V, E)$ where $E = \{\langle v_i, v_j\rangle | \forall v_i, v_j \in V\}$ ($*$ Initialize G as the complete undirected graph on V $*$)
2. $setSize \leftarrow -1$
3. **repeat**
4. $setSize \leftarrow setSize + 1$
5. **repeat**
6. Select a (new) ordered pair of variables (v_i, v_j) that are adjacent in G such that $|adj(G, v_i) \setminus \{v_j\}| \geq setSize$
7. **repeat**
8. Choose (new) $K \subseteq adj(G, v_i) \setminus \{v_j\}$ with $|K| = setSize$
9. **if** $(v_i \perp\!\!\!\perp v_j | K)$
10. **then** $E \leftarrow E \setminus \langle v_i, v_j\rangle$ ($*$ Delete edge $\langle v_i, v_j\rangle$ $*$)
11. $S(i, j) \leftarrow K$; $S(j, i) \leftarrow K$ ($*$ Store the separating set $*$)
12. **until** edge $\langle v_i, v_j\rangle$ is deleted or all $K \subseteq adj(G, v_i)\setminus\{v_j\}$ with $|K| = setSize$ have been chosen
13. **until** all ordered pairs of adjacent variables v_i and v_j such that $|adj(G, v_i) \setminus \{v_j\}| \geq l$ and $K \subseteq adj(G, v_i) \setminus \{v_j\}$ with $|K| = setSize$ have been tested for conditional independence
14. **until** for each ordered pair of adjacent nodes v_i, v_j, $|adj(G, v_i) \setminus \{v_j\}| < setSize$

Spirtes et al. provide the proof that this algorithm produces the correct skeleton in [SGS00]. The maximum value reached by the variable $setSize$ is determined by the maximum degree of the underlying graph, proved in [KB07]. The next stage in the PC

algorithm is to orient the edges.

Algorithm PC_{or}

Input: $G(V, E), S$

($*$ An undirected graph, set of separating sets $*$)

Output: $\tilde{G}(V, \tilde{E}, A)$

($*$ An partially oriented graph that represents the Markov equivalence class $*$)

1. $\tilde{G} \leftarrow G$; $A \leftarrow \emptyset$; $\tilde{E} \leftarrow E$
2. **for** all $\langle v_i, v_j \rangle \notin E$ such that $\exists v_k, \langle v_i, v_k \rangle \in E \wedge \langle v_j, v_k \rangle \in E$ ($*$ Rule 0 $*$)
3. **if** $v_k \notin S(i, j)$
4. **then** $\tilde{E} \leftarrow \tilde{E} \setminus \{\langle v_i, v_k \rangle, \langle v_j, v_k \rangle\}$
5. $A \leftarrow A \cup \{\langle \overrightarrow{v_i, v_k} \rangle, \langle \overrightarrow{v_j, v_k} \rangle\}$ ($*$ Orient v_i—v_k—v_j as $v_i \rightarrow v_k \leftarrow v_j$ $*$)
6. In the resulting CPDAG, repeatedly apply the following rules until no more rule can be applied.
7. R1: Orient v_i—v_j into $v_i \rightarrow v_j$ whenever there is an arc $v_k \rightarrow v_i$ such that v_k and v_j are nonadjacent.
8. R2: Orient v_i—v_j into $v_i \rightarrow v_j$ whenever there is a chain $v_i \rightarrow v_k \rightarrow v_j$.
9. R3: Orient v_i—v_j into $v_i \rightarrow v_j$ whenever there are two chains v_i—$v_k \rightarrow v_j$ and v_i—$v_l \rightarrow v_j$ such that v_k and v_l are nonadjacent.
10. R4: Orient v_i—v_j as $v_i \rightarrow v_j$ whenever there are two chains v_i—$v_k \rightarrow v_l$ and $v_k \rightarrow v_l \rightarrow v_j$ such that v_k and v_j are nonadjacent and v_i and v_l are adjacent.

As reported by Pearl [Pea00], the repeated application of a set of rules are sufficient to orient *all* arrows that are common to the equivalence class of the causal model [Mee95]. Rules 1 through 4 are also termed as the Meek orientation rules. If PC_{sk} outputs the correct graph and the correct separation sets, PC_{or} is simply a deterministic application of the *d*-separation criterion.

Rarely in practice is perfect conditional independence information available and one

has to infer these relationships from data samples. Hence, we are forced to relax the assumption that perfect conditional independence information is available to the algorithm PC_{sk}. Although PC_{or} does not need conditional independence information directly, it indirectly uses this information reflected both in the structure of the skeleton and in the content of the separation sets S. In the next section we discuss a method for determining conditional independence from data.

3.2 Determining Conditional Independence

In this thesis I assume that all the variables are multinomial random variables and I focus on determining conditional independence based on this assumption. Mutual Information of two random variables is a quantity that measures the mutual dependence of the two variables. Formally mutual information can be defined as follows:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) \log \left(\frac{P(x,y)}{P(x)P(y)} \right) \tag{3.1}$$

Mutual information is usually measured in bits, and thus the logarithmic term is of base 2. Mutual information quantifies the dependence between the joint distribution of X and Y, $P(X,Y)$ and what the joint distribution would be if X and Y were independent. $I(X,Y) = 0$ if and only if X and Y are independent random variables; the logarithm term vanishes if for every $x \in X$ and $y \in Y$, $p(x,y) = p(x)p(y)$.

Intuitively, mutual information measures the information that is shared by X and Y. It is a measure of how much the knowledge of one of these variables reduces our uncertainty about the other. For example, if X and Y are independent, then knowledge about X does not diminish the uncertainty we have about Y by any amount and vice versa; therefore we say that their mutual information is zero. On the other hand, if X and Y are identical

then knowledge about X gives us identical knowledge about Y, and vice versa. As a result, in the case of identity the mutual information is the same as the uncertainty in Y (or X) alone, which can also be termed as the entropy of Y (or X), where entropy $H(X) = -\sum_{x \in X} P(x)log(P(x))$.

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) \ - H(X|Y) - H(Y|X) \end{aligned} \tag{3.2}$$

However, we are interested in a quantity that measures independence among two variables, given information about another set of variables. The analogous quantity that does this is called *conditional mutual information*, sometimes referred to as *conditional cross entropy*. In equation 3.2 we replace the entropy terms by terms conditioned on a third variable set Z: $I(X,Y|Z) = H(X|Z) - H(X|Y,Z)$. More formally, we can write conditional mutual information as:

$$I(X,Y|Z) = \sum_{z \in Z} P(z) \sum_{y \in Y} \sum_{x \in X} P(x,y|z) \log \left(\frac{P(x,y|z)}{P(x|z)P(y|z)} \right) \tag{3.3}$$

This value is *zero* only when there is perfect conditional independence among X and Y, given Z. To test for conditional independence from data sampled from a distribution, I first compute the maximum likelihood (ML) estimates of the probability terms in the above expression. I assume complete datasets, that is, every data item has valid values for each variable. If $X = \{X_1, \ldots, X_n\}$ is the set of variables, a complete dataset can be written as $D = \{D_1, \ldots D_m\}$. The maximum likelihood estimate for the probability of a variable is then

$$P(X = x_j)_{MLE} = \frac{N_j}{m} \tag{3.4}$$

where N_x is the number of data items in which X took the value x_j. ML estimates for joint distributions among more than one variable can be calculated similarly by treating the variables as one composite variable where their set of possible values is from their Cartesian product. For example, for binary variables X and Y the set of possible values are $\{00, 01, 10, 11\}$. For small sample set sizes (small m), the ML estimate can suffer from zero counts, i.e. some configurations of the variables may never appear in the data set. Typical methods used to correct for this are by using a Laplace correction or by setting a Dirichlet prior. The Laplace correction assumes that each of the data configurations has appeared at least once in the data set and starts the count from 1 instead of 0. This gives us a different estimate (P_{LC}) for the probability (equation 3.5).

$$P(X = x_j)_{LC} = \frac{N_j + 1}{m + r} \tag{3.5}$$

where r is the arity of X. Note that $m = \sum_{j=1}^{r} N_j$. This has however been criticized as biasing the estimate "too much" towards the uniform distribution. A refinement that is commonly used among Bayesian practitioners is to use a Dirichlet prior with parameters $\alpha = \{\alpha_1, \ldots, \alpha_r\}$. The parameters of the Dirichlet prior (or simply Dirichlet parameters) can be thought of as pseudo-counts that represent our prior belief about the distribution, as can been seen in equation 3.6.

$$P(X = x_j)_{Dir} = \frac{N_j + \alpha_j}{m + \alpha_0}, \quad \text{where } \alpha_0 = \sum_{j=1}^{r} \alpha_j \tag{3.6}$$

Note that as sample size m increases the effect of the priors vanish and they all converge on the ML estimate. As suggested by Heckerman [Hec96, Hec99] I use a uniform conjugate Dirichlet prior with $\alpha_j = 1/r$.

The test statistic used for independence is G^2, which is $2mI(X, Y|Z)$ where m is the sample size (equation 3.7). Under assumptions of independence, it is known that G^2 follows a χ^2 distribution (equation 3.7) with γ degrees of freedom (equation 3.8) [Fis29].

$$G^2_{test} = \chi^2_{test} = 2mI(X, Y|Z) \tag{3.7}$$

$$\gamma = (\gamma_x - 1)(\gamma_y - 1)\prod_{a \in Z} \gamma_a \tag{3.8}$$

The χ^2 distribution has the following probability distribution function for γ degrees of freedom.

$$\chi^2_\gamma = \frac{1}{2^{\gamma/2}\Gamma(\gamma/2)} x^{\gamma/2-1} e^{-x/2} \quad , x \in [0, \infty) \tag{3.9}$$

The Γ function is an extension of the factorial function to real and complex numbers, and has the property $\Gamma(n) = n\Gamma(n-1)$ for all real and complex values with non-negative real parts. It has closed form values for half-integers, and since γ is always an integer in our problem, it is computable, and

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \tag{3.10}$$

The probability Q that a χ^2 value calculated for an experiment with γ degrees of freedom is due to chance is

$$Q_{\chi^2,\gamma} = \frac{1}{2^{\gamma/2}\Gamma(\gamma/2)} \int_{\chi^2}^\infty x^{\gamma/2-1} e^{-x/2} dx \tag{3.11}$$

I now use the ubiquitously used Pearson's chi-squared test to determine whether a *null hypothesis* stating that the distribution of certain events observed in a sample is consistent with a particular theoretical distribution [Pea04]. When applied to a test for independence, the null hypothesis is that the observations, consisting of the values of two outcomes, are statistically independent. For the test of independence, the χ^2 probability can be calculated by using a look-up table or by a fast approximate numeric integration technique since there is no closed form for equation 3.11. If the χ^2 probability is less than or equal to the chosen value of parameter α, known as the *significance value*, (or the X^2 statistic is larger than the critical point), we reject the null hypothesis of independence. In other words, when the test of independence succeeds, we can declare the conditional independence statement $(X \perp\!\!\!\perp Y|Z)$ true. It is common practice among statisticians to choose α to be 0.05.

3.3 Problems with the PC algorithm

Recall that the PC algorithm assumes *faithfulness*, i.e. the independence relationships among the variables are exactly those represented in the causal model and the d-separation criterion [Pea88]. When we have to rely on finite sample data sets and a statistical methods to infer conditional independence, we stand the risk of violating the faithfulness assumption. An example is shown in Figure 2.5. A similar example shown by [RZS06], is as follows. Consider the causal graph $A \rightarrow B \rightarrow C$, where $(A \perp\!\!\!\perp B|C)$ as well as $(A \perp\!\!\!\perp C)$. We can think of the second independence statement becoming true because B *cancels* A's direct effect on C. The PC algorithm, would find that $(A \perp\!\!\!\perp C)$, remove the edge $A - C$ and record $\emptyset$ as the separating set S_{AC}. In the orientation stage, this would result in the following incorrect result $A \rightarrow B \leftarrow C$ as $B \notin S_{AC}$. Note however, that due to the special nature of the unfaithfulness of this example, the algorithm finds the right skeleton anyway. This motivates the division of the faithfulness assumption into two separate implications, *Adjacency-Faithfulness* and *Orientation-Faithfulness*.

Definition 3.3.1 (Adjacency Faithfulness)**.** *Given a set of variables V whose causal structure can be represented by a DAG G, if two variables X, Y are adjacent in G, then they are dependent conditional on any subset of $V \setminus \{X, Y\}$.*

This definition emphasizes that two variables are non-adjacent in G if and only if no separating set is found. The second part, deals with *unshielded triples*, i.e. a set of three variables that has exactly one non-adjacency.

Definition 3.3.2 (Orientation Faithfulness)**.** *Given a set of variables V whose causal structure can be represented by a DAG G, let $\langle X, Y, Z\rangle$ be any unshielded triple in G.*

1. *If $X \rightarrow Y \leftarrow Z$ then X and Z are dependent given any subset of $V \setminus \{X, Y\}$ that contains Y.*
2. *Otherwise X and Z are dependent conditional on any subset of $V \setminus \{X, Y\}$ that does not contain Y.*

Orientation faithfulness specifies the conditions for the presence or absence of unshielded *v*-structures (also called unshielded colliders). For the unshielded triple $\langle X, Y, Z\rangle$ to exist, it only requires that a conditioning set is found, but for the orientations to be directed towards Y we require information that Y was not part of that set, or any other set that renders X independent of Y. In practice, we find that statistical tests of independence are more robust with respect to adjacency faithfulness than with respect to orientation faithfulness, as is expected. Orientation faithfulness imposes a stronger and hence more statistically error prone constraint than adjacency faithfulness.

The Conservative PC (CPC) algorithm by Ramsey et. al. [RZS06] relaxes the faithfulness assumption by assuming only adjacency faithfulness but attempts to verify orientation faithfulness to the extent possible. Previously, the SGS algorithm by Spirtes et.al. [SGS00] checks for the unshielded collider and non-collider condition, but barely fails short of correctness as it does not check for *unfaithful* unshielded triples that fail both these condition.

Additionally, the SGS algorithm is terribly inefficient, best case exponential, as it checks for dependence between X and Z conditional on every subset of $V \setminus \{X, Z\}$. CPC_{or} does better, as it uses the same intuitions as the PC algorithm by checking only for subsets that include potential parents of X and Z. The CPC algorithm replaces collider orienting lines 3, 4 and 5 in PC_{or}, with checks that mimic the orientation faithfulness criterion.

Algorithm CPC_{or}
Input: An undirected graph G
Output: A partially oriented DAG with unfaithful triples marked
1. **for** all subsets of the potential parents of v_j and v_j
2. **if** v_k is NOT in any subset conditional on which v_i and v_j are independent
3. **then** Orient $\langle v_i, v_k, v_j \rangle$ as an unshielded collider
4. **if** v_k is in all subsets conditional on which v_i and v_j are independent
5. **then** Leave $\langle v_i, v_k, v_j \rangle$ as an unshielded triple
6. Otherwise mark the triple $\langle v_i, v_k, v_j \rangle$ as unfaithful

The output of the CPC algorithm is an extended pattern (e-pattern) that contains undirected and directed edges, as well as unshielded triples marked unfaithful. Note that e-patterns are no longer in the Markov equivalence class of DAGs, but they represent a larger class that includes the set of graphs that unfaithful distributions can entail. Proof of correctness under the adjacency-faithfulness assumption is provided in [RZS06]. The rest of the algorithm proceeds as the PC algorithm does by applying the orientation rules (R1 through R4) in PC_{or}, avoiding the unfaithful triples. If further orientation rules result in resolving some of the orientation triples, the unfaithfulness mark is removed.

3.4 The *soft*-CPC algorithm

In practice, the PC algorithm has a tendency to label unshielded triples as colliders too often, due to a single faulty CI test that appeared first in the order of CI tests conducted. The CPC_{or} algorithm, "conservative" as its name suggests, orients colliders only when the common neighbor is not present in all of the separating sets. Both approaches are correct, given their assumptions. While PC assumes perfect CISs with respect to faithfulness, CPC_{or} assumes perfect CISs with respect to adjacency faithfulness. It has been noted that additional adjacency checks for unshielded triples could help resolve some of these problems. Additionally, they also refrain from incorporating any sort of tolerance for unfaithful triples detected due to faulty CISs. The CPC_{or} algorithm performs two extremely constrained tests, namely the membership or non-membership of a common neighbor in all separating sets. Suppose that there are a large number of separation sets for a non-adjacent pair, and one of the common neighbor is a member in only one of these sets. In CI tests on finite samples, it is possible that the single test that makes the condition for unshielded colliders fail is faulty, rather than that this be a real example of unfaithfulness. The alternate condition is also symptomatic: if it is absent only in one separation set.

As pointed out elsewhere, we reiterate here that unfaithfulness is a highly unlikely occurrence in many causal models [Pea00, SGS01]. It is parameter dependent and requires the precise tuning of several parameters to manifest itself. Arguments that justify their occurrence rely on the presence of some meta-mechanisms exogenous to the causal model that regulate and tune these parameters. Some examples of unfaithfulness inducing meta mechanisms could be evolution, other kinds of ecological balance mechanisms, market self regulatory phenomenon in economics, etc. Discounting for these sort of mechanisms, it is far more likely that the PC_{sk} algorithm errs in the adjacency detection stage, which will lead to an incorrect inference about unfaithfulness in triples adjacent to that error.

Unfaithful markings in e-patterns are a cumbersome weight to carry around in causal

models. They represent a larger family of DAGs than the CPDAGs produced by the PC algorithm and it is worth marking unfaithfulness only when we are confident about the detected unfaithfulness.

Additionally, performance of the CPC algorithm is harder to evaluate. Simple metrics based on edge comparisons are not sufficient as one needs prior knowledge about the unfaithful triples in the true causal graph. Even if this is available for validation tests, it requires meticulous parameter tuning when we generate test graphs. Even when we have access to the true distributions, verifying the unfaithfulness of an unshielded triple requires a series of Bayesian inference queries, which is NP-hard [Coo90, Pea00, WB05].

In practical applications, the prior knowledge about unfaithful triples is almost never available, needed for validating the results of CPC_{or}. Further, the additional information about an unfaithful unshielded triple is that it could potentially be a collider, whereas a faithful triple cannot. In the context of using future experimental data, this difference is not significant, as experimental data almost always supplies us with superior orientation information, as we shall see in sections 3.5 and 3.6. However, CPC_{or} reduces the number wrongly marked unshielded colliders and this is a valuable improvement that we would like to incorporate. I propose a simple modification to the PC algorithm which we call the *soft*-CPC algorithm (sCPC) that reduces the number of unfaithful triples by a quantification of unfaithfulness.

My contribution is to introduce a tunable parameter called *unfaithfulness tolerance* (β) that supports the determination of the level of conservativeness that is applied to constraint based search algorithms. The $sCPC_{or}$ algorithm replaces the CPC_{or} algorithm with the following lines.

Algorithm $sCPC_{or}$

Input: An undirected graph G, tolerance parameter β

Output: A partially oriented DAG with unfaithful triples marked

1. **for** all subsets of the potential parents of v_j and v_j
2. **if** v_k is NOT in at least a β fraction of subsets conditional on which v_i and v_j are independent
3. **then** Orient $\langle v_i, v_k, v_j \rangle$ as an unshielded collider
4. **if** v_k is in at least a β fraction of all subsets conditional on which v_i and v_j are independent
5. **then** Leave $\langle v_i, v_k, v_j \rangle$ as an unshielded triple
6. Otherwise mark the triple $\langle v_i, v_k, v_j \rangle$ as unfaithful

Additionally, we also propose the $PC_{minSepSet}$ algorithm, a simple but less greedy modification to the PC algorithm. The difference between $PC_{minSepSet}$ and PC is that when testing for conditional independence among sets of size k, the PC algorithm chooses the first separating set that it finds, while $PC_{minSepSet}$ continues CI testing for all sets of size k and stores all the separating sets for each pair of non adjacent variables in increasing order of conditional mutual information. Hence, $PC_{minSepSet}$ performs a larger number of CI tests (on average) than PC but the same in the worst case. $PC_{minSepSet}$ simply uses these stored separating sets for the orientation state and no new CI tests are evaluated unlike CPC and sCPC(β) versions.

We will discuss the comparative performance of all these algorithms in section 4.2.

3.5 Parental Search Algorithm

The PC family of constraint based algorithms represent the limit of what can be learned from static distributions or observational data alone. In this section we explore an algorithm that combines information from both static and an interventional distributions. We examine perfect interventions on multiple variables, that *set* variables to fixed values. We derive inspiration for these assumptions from related research that exploits notion of contexts in graphical models. Context partitioned models have been used to reduce the complexity of representation and inference by including only the most relevant stochastic variables, and pruning away the contextual variables[SLS07]. In this way, we are able to store more concise contextual models in a context model library, a more descriptive form or representing a dynamic stochastic system [SRLS08].

An operational context is a period of stability in the dynamic behavior of the system where a subset of the observed variables remain at fixed values. The contextual variables define the current context by remaining instantiated at a single point throughout the duration of the context while the stochastic variables maintain a stable joint distribution during that context [SLS07, SRLS08].

Let a set C be the set of *contextual* variables and $V \setminus C$, the set of *stochastic* variables. The assignment $C = c$, is a context and represents a hard intervention. Note that the set C is not unique to a context but can be identical for several contexts ($2^{|C|}$ of them).

Contexts are Interventions The learning agent may not always have interventional control over the quantities it measures, nor is it realistic that it is given this information explicitly at all times. It is desirable that it has the ability to infer that these interventions have occurred entirely from observations. There has been some work in the Bayesian learning community on detecting the targets of interventions [EM07], but an analogous method is not available for constraint-based methods in the general case. Although this is a very intriguing related area [KBDG04, LJY07] methods to detect interventions in real-time are

not in the scope of this dissertation. However, contexts, or hard interventions, are easy to detect. In this dissertation, any variable that takes a single value throughout the data set is determined to be a contextual variable.

A Context Transition occurs in a system when the interventional state is changed from $C = c$ into any of the other states of the system $C_x = c_x$. In lieu of these definitions, the *null* context represents a stable distribution obtained under no interventions.

Recall that joint distribution P, represented by a causal Bayesian network factorizes as the product of the conditional distributions of its variables given its parents (Equation 2.4). Let $C \subseteq X$ be the set of variables on which the intervention is performed (typically $|C| << |X|$). Let $\hat{c}'_j$ represent the values taken by variable $x_j \in C$. Then $P_{\hat{c}}$ is the post-interventional joint distribution for a particular context which we can also represent as $P(X|do(C = c))$ or $P(x_i, \ldots, x_n|\hat{c}'_i)$. Generalizing Equation 2.8 to multiple interventional variables we have the following:

$$P(x_i, \ldots, x_n|\hat{c}'_i) = \begin{cases} \prod\limits_{x_j \notin C} P(x_j|pa_j) & \text{if } x_j = \hat{c}'_i, \quad \forall x_i \in C \\ 0 & \text{if } x_i \neq \hat{c}'_i \end{cases} \tag{3.12}$$

From equation 3.12 we get the relationship between the pre-interventional and multivariable post-interventional distributions as follows.

$$P_{\hat{c}} = \frac{P}{\prod\limits_{x_j \in C_i} P(x_j|pa_j)} \text{ whenever } x_j = \hat{c}'_i, \quad \forall x_i \in C \tag{3.13}$$

Note that we have the right hand side of equation 3.13 entirely in factors of the pre-interventional distribution P. Conditional probability factors corresponding to the context set C increases the density on P in specific locations. Suppose we have sufficient samples

to estimate P and its relevant factors, the right hand side of equation 3.13 can be calculated purely from the pre-interventional distribution given that we know pa_j. We denote an estimate obtained such as $P'_{\hat{c}}(pa_j)$. $P_{\hat{c}}$ can also be estimated from the post-interventional distribution, and this should match $P'_{\hat{c}}(pa_j)$, given that we know the unknown parent set pa_j for the contextual variables $x_j \in C$. We turn this search problem, of finding a set $\hat{pa}_j$ that satisfies the above relationship, into a minimization problem.

$$\hat{pa}_i = \underset{pa_j \subseteq 2^{X \setminus \{x_i\}}}{\textbf{argmin}} \; D(P_{\hat{c}}, P'_{\hat{c}}(pa_j)) | x_j \in C \tag{3.14}$$

where $D()$ is a suitable measure of divergence between two distributions

The search for pa_j in the powerset of $X \setminus \{x_j\}$ looks worrisome, but we can reduce the complexity of this search by using several methods. First, we restrict the search to the Markov blanket (set of neighbors of a node in the skeleton) for each variable or restrict the cardinality of the parent set to a maximum for graphs of known finite degree. Even better, we can use P as input to the PC algorithm described earlier in this chapter, and use the PDAG to constrain the potential set of parents. Potential parent sets can be restricted to those that contain nodes that are already determined as parents by the PC algorithm, unioned with each subset of the set of adjacent nodes that are yet unoriented.

When such a set pa_i is found, we can orient each edge adjacent to all $x_j \in C$ and then update the PDAG.

Algorithm *parentalSearch*($P, P_{\hat{c}}$)

Input: $P, P_{\hat{c}}$

($*$ Pre and post Interventional distributions $*$)

Output: $pa_{\hat{j}}|x_j \in C_i, H'(P)$

($*$ parent set for each contextual variable, augmented PDAG $*$)

1. $H(P) \leftarrow PC(P)$

($*$ A partially oriented DAG compatible with P $*$)

2. **for** $x_j \in C$
3. Compute $pa_{i,H}$, the set of parents of x_j in $H(P)$
4. Compute $nb_{i,H}$, the set of undirected neighbors x_j in $H(P)$
   for $n_i \in 2^{nb_{j,H}}$
5. $pa_j \leftarrow pa_{j,H} \cup n_i$
6. Compute difference metric d_{pa_j}
7. $pa_{\hat{i}} \leftarrow \underset{pa_i}{\textbf{argmin}}\, d_{pa_i}$
8. Update $H(P)$ with orientations $\langle pa_{\hat{j}}, x_j \rangle$ and $\langle x_j, nb_j \setminus pa_{\hat{j}} \rangle$ for each $x_j \in C$.
9. In the resulting PDAG apply rules R1 through R4 as specified in PC_{or}, in lines 7,8,9 & 10.

Note that the number of variables chosen for intervention for each context affects the size of the search space. However, with the constraints specified by $H(P)$, we expect that this number is not too large when considering contexts that set two or three variables at a time. Importantly, note that contexts of size 2 can be especially useful, considering the unfaithfulness problem discussed in section 3.3. Using *parentalSearch*, unfaithful unshielded triples can be oriented by measuring the distribution entailed by subjecting the two non-adjacent nodes in the triple to an intervention. Suppose the maximum degree for all variables computed from $H(P)$ is d, then the total size of the search space is 2^{dk} when contexts are set with k variables at a time. Moreover, as $H(P)$ is updated at each context transition, the number of undirected neighbors reduces by at least one for each of

the parents of the contextual variables, potentially halving the search spaces involved in future contexts. Therefore, we can expect the algorithm to perform reasonably in practical domains. I present the results of this algorithm in section 4.3.

Note that we did not specify the divergence metric d_{pa_i} that is to be used. Commonly, the Kullback-Leibler (KL) divergence [KL51] is a popular measure that is used to measure the information loss between two distributions. Given that the dimensionality of the distributions can be arbitrarily high, KL-divergence cannot be measured directly. In the next section, I describe an algorithm that computes the approximate KL-Divergence from data sampled from two distributions.

3.5.1 Approximate Kullback-Leibler Divergence

The Kullback-Leibler divergence is a non-symmetric measure of the difference between two probability distributions P and Q. It measures the expected number of bits required to code samples from P using a code based on Q, rather than using a code based on P. Typically, P represents the "true" distribution and the KL-divergence measures how different the "approximation" Q, is from P. If P and Q are defined over discrete random variables, then the KL-divergence is defined as :

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \tag{3.15}$$

We consider the problem of computing the KL-divergence of two distributions when we are given sparse datasets sampled from each of the distributions. Intuition suggests that we simply compute the ML (maximum likelihood) estimates of each distribution and compute the summation in equation 3.15. However, the distributions can be of arbitrarily high dimension, N. A distribution that is represented by an N node Bayesian network with binary nodes has 2^N possible configurations. Such a large distribution cannot be stored, which is precisely one of the utilities of using a Bayesian network. We have to assume

that we are unaware of the Bayesian network structure, as we intend to use this measure in the process of causal discovery. Alternatively, we can assume a Bayesian network, or construct one that is compatible with the Markov equivalent class of networks determined by the IC/PC algorithm. However, the summation in 3.15 is still cripplingly slow; the loop is 2^N long and the time complexity is exponential.

I propose a simple alternative in the following algorithm *approximateKLD*, that enjoys a polynomial time complexity with respect to the size of the dataset m and the dimensionality N. Therefore, each dataset can be considered a set (or vector) of m data samples where each data sample is a vector of length N. Note that each data sample can also be thought of as an index into its respective distribution.

Algorithm *approximateKLD*(D_1, D_2)

Input: D_1, D_2

($*$ Two i.i.d. data sets from two distributions of the same dimensionality. $*$)

Output: KLD_{app}

($*$ Approximate KL-divergence between D_1 and D_2 $*$)

1. $D_1' \leftarrow sort_{lex}(D_1)$
2. $D_2' \leftarrow sort_{lex}(D_2)$ ($*$ Sort both the data sets in lexicographical order $*$)
3. $m_1 \leftarrow length(D_1')$; $m_2 \leftarrow length(D_2')$;
4. $currentIndex \leftarrow min(D_1'(1), D_2'(1))$ ($*$ $D(i)$ returns the i^{th} data sample in dataset D $*$)
5. $i \leftarrow 1$, $j \leftarrow 1$
6. $KLD_{app} \leftarrow 0$
7. $N_1 = 0$, $Nq = 0$
8. **while** $(i \leq m_1 \vee j \leq m_2)$
9. **do**
10. **if** $(currentIndex = D_1(i) \vee currentIndex = D_2(i))$
11. **then if** $(D_1(i) = D_2(i))$
12. **then** $N_1 \leftarrow N_1 + 1$, $i \leftarrow i + 1$, $N_2 \leftarrow N_2 + 1$, $j \leftarrow j + 1$
13. **else if** $(D_1(i) < D_2(i))$
14. **then** $N_1 \leftarrow N_1 + 1$, $i \leftarrow i + 1$
15. **else** $N_2 \leftarrow N_2 + 1$, $j \leftarrow j + 1$
16. **else**
$KLD_{app} \leftarrow KLD_{app} + \frac{N_1+\alpha_k}{m_1+\alpha_0} \log \frac{(N_1+\alpha_k)(m_2+\alpha_0)}{(N_2+\alpha_k)(m_1+\alpha_0)}$
17.
18. $N_1 \leftarrow 0$, $N_2 \leftarrow 0$
19. $currentIndex \leftarrow min(D_1'(1), D_2'(1))$

The two lexicographical sorts in lines 1 and 2, have complexity $O(mN \log m)$ each (as-

suming equally long datasets). The N term is due to the fact that the comparison operator has a worst case of $O(N)$, when comparing two data vectors of length N. The rest of the algorithm can be likened to the *merge* operation of merge sort, where we co-iterate through two sorted lists, but instead of merging the two lists, we aggregate the component of the KLD in a sum. Co-iteration has a complexity $O(m_1 + m_2)$ or rather, $O(m)$ for datasets of comparable sizes and is dominated by the two sorts. Therefore, this algorithm has a time complexity of $O(Nm \log m)$ in the worst case, which is a tractable quantity.Thus, we can compute an approximate measure of distance between two distributions given only their datasets.

Note that in line 16 of the algorithm, I use a Bayesian estimate with a Dirichlet prior with the uniform parameters α_k for all configurations. We expect that for very small sample sizes, the KLD_{app} will deviate from the true KL-divergence but in the large sample limit ($m >> 2^N$) the summation converges on the true KL-divergence. We only intend to use this measure to measure the relative difference of two distributions with respect to a third distribution, and we find that this method works well in practice. Some exemplary results can be found in section 4.4.

3.6 Interactive Causal Discovery

In the previous section I presented an algorithm to learn causal relationships from hard interventions on multiple variables by finding the most likely set of parents of the intervened variables. In this section I describe an interactive algorithm that can be used to discover causal relationships from a series observational and interventional (hard or soft) datasets. Soft interventions are also termed as "parametric interventions" [ES06] or "mechanism changes" [TP01a]. In section 2.5, Equation 2.9 describes the relationship between the pre- and post-interventional joint distributions with respect to a parametric intervention performed on a single focal variable $x_j \in X$. We noted that the potential parents of the

focal variable could change from pa_i to pa_i^*, thereby implying a potential change in the causal graph. In this section we assume that the causal graph is static and does not change with respect to parametric interventions following [ES06]. If P and P^* are the pre- and post-interventional distributions respectively, and a parametric change is performed on a single focal variable x_i, we have the following.

$$P^*(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)\frac{P^*(x_i|pa_i)}{P(x_i|pa_i)} \tag{3.16}$$

In 2001, Tian and Pearl show that by carrying out a series of single variable parametric interventions [TP01a], N of them, and by testing only for marginal changes, they can detect the hierarchy of descendants in the causal graph. Consequently, if this hierarchy can be used as additional constraints in the PC algorithm, one can recover the complete causal graph. In the empirical part of their study, Tian and Pearl collect a large number of samples for each distribution representing the parametric interventions as a controlled perturbation of the original parameters.

Later in 2006, Eberhardt [ES06] provided the following theoretical result that proved to be a significant advancement to the idea. The idea involves augmenting the causal graph with interventional nodes and then deploying the basic idea of the PC algorithm to establish the orientations of the nodes upon which interventions have been performed. Unlike ordinary nodes, interventional nodes can have only outward arcs, and this helps us establish not only the unshielded colliders associated with the interventional nodes, but also the orientation of the unshielded triples. This leads to some remarkable results and provides great insight into the experimental methodology that is to be used in causal learning. For clarity, I reproduce this important theorem and proof by Eberhardt here.

Theorem 3.6.1. *One experiment is sufficient and necessary to discover the causal structure among a set of N causally sufficient variables if multiple variables can be independently and simultaneously subjected to a parametric intervention per experiment.*

The proof given in [ES06] can be summarized as follows:

Proof. Given faithfulness, observational data alone is sufficient to determine adjacencies. Then, consider that we augment the causal structure among variables $X = \{x_i \dots x_N\}$ with the intervention nodes $I = \{I_i \dots I_N\}$, where each intervention node I_i represents an independent parametric change on x_i. Each I_i has exactly one outward edge directed towards x_i. Given the distribution entailed by an experiment represented by this augmented graph, we can find the separation set that obtains the conditional independence $(I_i \perp\!\!\!\perp x_j | S)$, where x_j is a node adjacent to x_i. If $x_i \notin S$, then $\langle I_i, x_i, x_j \rangle$ form an unshielded collider at x_i and we obtain the edge $\langle x_j, x_i \rangle$. Otherwise, x_i blocks the path from I_i to x_j and since we already know the orientation $\langle I_i, x_i \rangle$, this implies the edge $\langle x_i, x_j \rangle$. □

The result is to be understood as a worst case analysis and it corresponds to the class of fully connected causal graphs. Due to the absence of any unshielded colliders in complete graphs, the orientation phase of the PC algorithm is unable to orient any edge. However, knowledge of the worst case result is very encouraging. In cases better than worse case, while the single experiment result still holds, we can do much better than N in the number of variables chosen for intervention. All N intervention nodes in I are not necessary as some of the orientations are already available from the PC family of algorithms run on observational data alone.

Let us discuss the general case. It has been suggested [Ebe08] that the key difficulty in uniquely identifying the causal structure is to determine the orientation of the edges that belong to cliques. Cliques are subsets of the vertex set for which every pair of vertices is connected by an edge in the true causal structure. A clique makes edge-orientations maximally independent, because fewer orientations are implied (absence of v-structures) and the only constraints are the acyclicity constraints. In other words, each clique acts like a worst-case subgraph. For an algorithm to minimize the number of experiments it has

to break down the cliques and find orientations in cliques as fast as possible. Eberhardt conjectures that the worst case number of experiments needed to fully orient a causal graph is $\lceil \log_2(|C_{max}|) \rceil$ where $|C_{max}|$ is the size of the largest clique in the graph. To support this conjecture, he also suggests an algorithm based on finding maximal cliques and provides experimental results on graphs up to 12 nodes [Ebe08]. However, a drawback of this algorithm is that finding maximal cliques is NP-complete and the algorithm does not scale well for larger graphs.

For graphs of a certain maximum degree, say k clique sizes are limited to k. Finding a k-clique has complexity $O(n^k k^2)$ for graph sizes of n. Although we can expect k to be small for a large class of graphs, this is still a limiting factor for algorithms that employ k-clique finding. We need a faster alternative of scoring vertices for experimental priority, depending on the number of small cliques they belong to.

Prior to Eberhardt's conjectured bound, Meganck et. al. suggest another method based on decision theory [MM06], to establish a scoring function that takes into account the possible number of edges that may become oriented due to an experiment. In their of experiments, they present the results of their algorithm on a simple, but interesting example of a 5 node causal model. They show that in some cases, the nodes with the smallest degree, and hence most unintuitive, may in fact be the best ones to be chosen for intervention. In the example, a node with degree 1, orients the maximum number of edges [MM06]. However, the decision theoretic approach is also very expensive even for small graphs. For each possible choice of intervened node, one has to enumerate all possible orientation configurations of its neighbors, and recursively explore all subsequent orientations to the full depth. The DAG members of the equivalence class of CPDAGs is in general unknown and is hard to compute. A scoring function on these DAGs is also very hard to formalize and the authors resort to an approximation and add in other heuristics like experimental costs and expert opinion to deal with the problem.

The works described above address either the worst case, or cases where intuitive

methods fail. Hence, they resort to sophisticated and expensive formulations to score the nodes. My hypothesis is that, in the average case, the problem of choosing interventional nodes is not that hard. In practice, we can choose an interventional node based on a simple and fast heuristic. In this section, we propose such a heuristic, which is intuitive and computationally inexpensive.

Our heuristic approach is simple, at each stage we evaluate the undirected degree of each node and assign it as the score. Nodes with higher scores have a higher priority for intervention. For each connected subgraph, clique nodes would get a node score of at least their clique size, so larger cliques would get priority over smaller cliques and lone undirected edges (2-cliques) would get the lowest scores in the clique size hierarchy. Therefore, this is compatible with Eberhardt's conjecture. Nodes with a small clique size but with high undirected degree and these will get a high priority. I believe that in the average case, orientations that are determined by an intervention on these nodes have a high probability of resulting in subsequent "free" orientations. In essence, this method is a first degree approximation of the decision theoretic approach while also accounting for the conjecture about clique-size. With this method, node intervention priorities can be initialized very quickly (after PC_{or}), in $O(E\log(E))$ time. Subsequent updates to the node scores would only require $O(E\log(E_{new}))$ where E_{new} is the number of orientations found after each intervention.

An automated causal learning algorithm, could simply choose the interventions based on this priority score. Alternatively, an interactive algorithm could provide this list to a human experimenter and allow her to make a choice based on expert judgment, experimental costs and other considerations. We provide below, the *interactiveCausalDiscovery* algorithm that can be used as a framework for interactive or automatic causal discovery.

Algorithm *interactiveCausalDiscovery*(P)

Input: P

($*$ Pre-interventional distribution $*$)

Output: $G(V, E)$

($*$ a fully oriented causal DAG $*$)

1. $H(P) \leftarrow PC(P)$

($*$ A partially oriented DAG compatible with P $*$)

2. Assign to each node a score equal to the number of undirected edges it is connected to.
3. **while** There is at least one undirected edge in $H(P)$
4. Suggest a set of interventions in descending order of the node scores.
5. Accept the experimental data, estimate the experimental distribution $P*$, and apply the PC algorithm to $H(P)$ with the new information $P*$, (including orientation rules R1 through R4)
6. Update the node scores corresponding to the new orientations.

The elegance of this algorithm lies in the observation that we are able to prompt the experimenter on the order of experiments she should carry out, with the highest priority given to the most difficult graph motifs (cliques and high degree nodes). The intervened node always disappears from priority list, and forced orientations (due to the Meek orientation rules [Mee95]) may cascade into further orientations among adjoining nodes. Some of the low priority nodes may also disappear from the next iteration of choices or may get rearranged in the node priority list. If the experimenter is able to perform a single experiment intervening on all the suggested variables, the algorithm terminates in one iteration. Given practical, ethical or other constraints, if the experimenter is able to perform only a subset of the interventions suggested, one may still be able to uncover the entire causal graph.

If the ubiquitously used ALARM network [BSCC89] is an indicative example of a typ-

ical causal graph, in terms of the number and size of cliques, average and maximum degree of the nodes, then a majority of typical causal graphs will require very few interventional experiments towards full causal discovery and simple heuristics such as suggested in *interactiveCausalDiscovery* will be sufficient to restrict the number of required experiments.

In section 4.5, we provide some very interesting results obtained on an empirical study of the above algorithm on sparse graphs like the ALARM network. We perform several tests varying several network parameters such as network size and complexity to demonstrate their effects on the number of experiments required.

Finally, in chapter 4, I present the results of using all three algorithms, at different stages in the task of causal discovery on several datasets. The results of my approach are compared to the results of the traditional approaches on each task.

Chapter 4

Experiments, Results and Discussion

"However beautiful the strategy, you should occasionally look at the results." - Sir Winston Churchill (1874 - 1965)

4.1 The Experimental Framework

This chapter begins by describing the experimental setup, including popularly used causal models, the generation of new and random models and the process of generating both observational and experimental data from these models. In Section 4.1.1, I describe the ALARM network, and in the subsequent section 4.1.2, how I generate several other networks of varying complexity and size. Section 4.1.3 presents the simple recursive sampling algorithm used in our tests.

In the next three sections 4.2, 4.3 and 4.5, I describe the experiments and results obtained on the three algorithms for causal learning we proposed in chapter 3. Section 4.3 also contains independent test results for the approximate KL-Divergence algorithm described in 3.5.1. Next, I provide a demonstration of the sequence of steps in causal discovery from the initial skeleton construction stage to full recovery of the causal graph applied

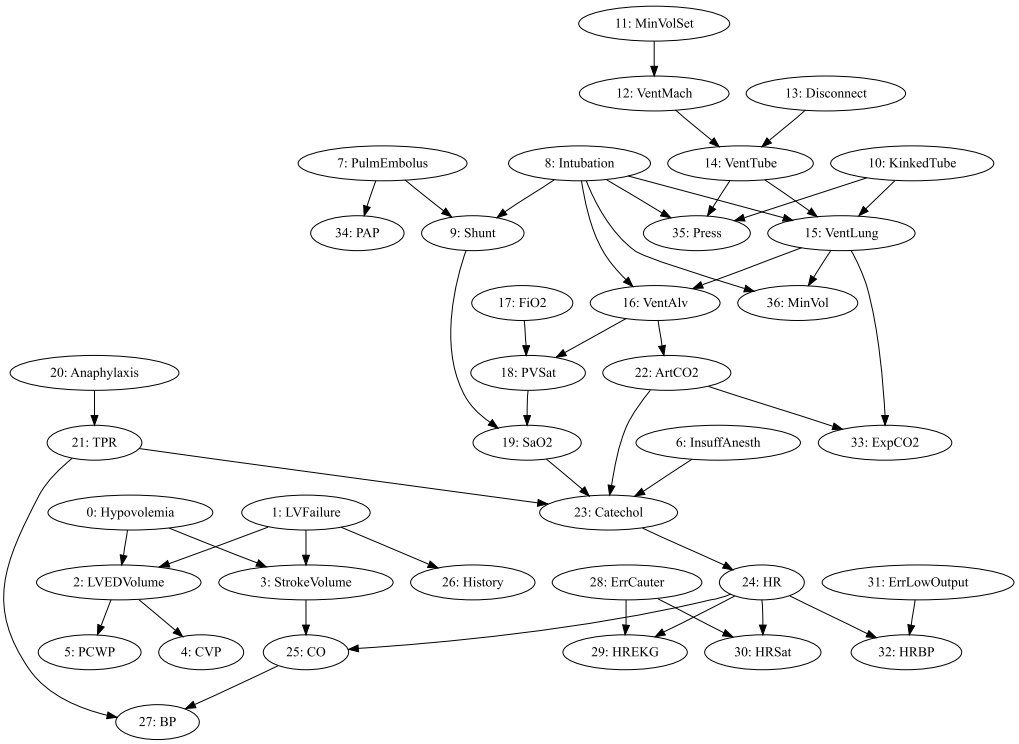


Figure 4.1: The ALARM causal Bayesian network

on a typical example with 50-nodes. Finally, Section 4.6 presents the indicative results of applying a constraint based causal learning algorithm on the Aircraft network (Fig. 1.2) modeled as an SEM.

4.1.1 The ALARM network

The ALARM network is a non-trivial Bayesian network which was first developed by medical experts for monitoring patients in intensive care [BSCC89]. As shown in figure 4.1, it has 37 nodes and 46 arcs with variable arities ranging from two to four. It is a very popular and well understood network in the field of Bayesian network inference and structure learning research, and used as a benchmark for testing several algorithms. The key to the popularity of the network is that it is of a reasonable size and complexity that

several of its features can be exactly computed, while at the same time it provides as a suitable challenge framework. We use the ALARM network with parameters (conditional probability tables) as provided by Norsys Software Corporation [Cor], as well as with modified parameters.

4.1.2 Random Causal Models

Apart from the ALARM network, I also carry out tests on a range of randomly generated causal Bayesian networks. I generate random DAGs with parameters for size, maximum degree and undirected average degree. I make use of the generic Boost graph library in C++ for the relevant data structures and algorithms in the implementation. My procedure for generating random causal models is as follows.

Algorithm *MakeRandomBayesNet*($N, maxDegree, avgDegree$)
Input: $N, maxDegree, avgDegree$
($*$ network size, maximum in-degree, average degree (undirected) $*$)
Output: $G(V, A, \Theta)$
($*$ Random Bayesian Network $*$)
1. $V \leftarrow \{v_1, v_2, \ldots, v_N\}$ ($*$ Set of nodes $*$)
2. $num_edges \leftarrow 0$
3. **while** $num_edges < (avgDegree \times N)$
4. **do** Choose a random pair of nodes $\langle v_i, v_j \rangle$ from V.
5. **if** $(inDegree(v_j) < maxDegree) \wedge v_j \not\longrightarrow v_i$ ($* \longrightarrow$ implies "reachable" $*$)
6. **then** Add arc $\langle v_i, v_j \rangle$ to A.
7. **for** each $v_i \in V$
8. **do for** each configuration of parents $pa_{i,j}$
9. **do** Assign uniformly random $\Theta_{i,j}$ and normalize.

In this manner, I am able to generate a large collection of causal models with a wide range of parameters to test the robustness of my algorithms. This supports the presentation of average performance of these algorithms across several different networks of comparable size and complexity.

In sections 2.5 and 3.6 I described parametric change or soft interventions. I implement this using two different methods:

1. For a given parameter change δ, I randomly choose a single probability parameter $\theta_{i,j,k} \in \Theta_{i,j}$. If $\theta_{i,j,k} < 0.5$, I add δ to it, otherwise subtract. Then the rest of the entries in $\Theta_{i,j}$ are re-normalized

2. Reassign new random values to $\Theta_{i,j}$.

4.1.3 Sampling

In this subsection I briefly describe the sampling algorithms used for generating data from causal Bayesian networks.

Algorithm *SampleBayesNet*(G, M)
Input: $G(V, A, \Theta), M$
($*$ Bayesian network, number of samples $*$)
Output: D_G
($*$ Set of samples $*$)
1. $D_G \leftarrow \{D_{1,1}, D_{1,2}, \ldots, D_{M,N}\}$ ($* N = |V| *$)
2. $D_{i,j} \leftarrow 0, \forall D_{i,j} \in D_G$ ($*$ valid multinomial samples are $> 0 *$)
3. **for** i from 1 to M
4. **do for** j from 1 to N
5. **do if** $D_{i,j} = 0$

6. **then** $D_{i,j} \leftarrow sampleVariable(v_i)$

The procedure $sampleVariable()$ in line 6 is defined recursively and proceeds as follows:

1. If sample $D_{i,j} > 0$ return $D_{i,j}$.

2. else if v_i is a root node, generate a random number r, in the interval $(0, 1)$ and find the first bin b in v_i's marginal distribution, such that the cumulative distribution of bin b is lesser than r. Return b.

3. else make recursive calls to $sampleVariable()$ on each of the parents of v_i. Once parents are sampled choose the corresponding probability table specified by the parent's samples (instead of the marginal in step 2) and generate a sample similar to step 2.

For generating samples from interventional distributions (hard interventions), we simply ignore the $sampleVariable()$ procedure for the interventional variables, and use their "set" values instead. The rest of the variables are sampled as before. Sampling from soft interventions proceeds identically once the Bayesian network has been modified as described in the previous section.

4.2 Comparative Performance of sCPC

In this section we describe the results of the $sCPC_{or}$ in comparison to the PC (PC_{sk} + PC_{or}), $PC_{minSepSet}$ and CPC algorithms. In identifying the unshielded colliders, the PC is most greedy and the CPC is most conservative, while $PC_{minSepSet}$ and the $sCPC_{or}$ are ordered in between.

In the following set of figures (4.2 through 4.5, I present the performance metrics of $sCPC_{or}$ with 4 different β (unfaithfulness tolerance) parameter settings (0.2, 0.4, 0.6 and 0.8) and $PC_{minSepSet}$ against PC and CPC_{or}. We vary sample sizes for the datasets from 500 to 50000, noted on the abscissa of each chart. Each result shown is the mean performance of each algorithm over 10 different dataset samples. The error bars denote the standard deviation of these metrics across these datasets. Note that, in these charts, the abscissas are just treated as categories and are not to scale.

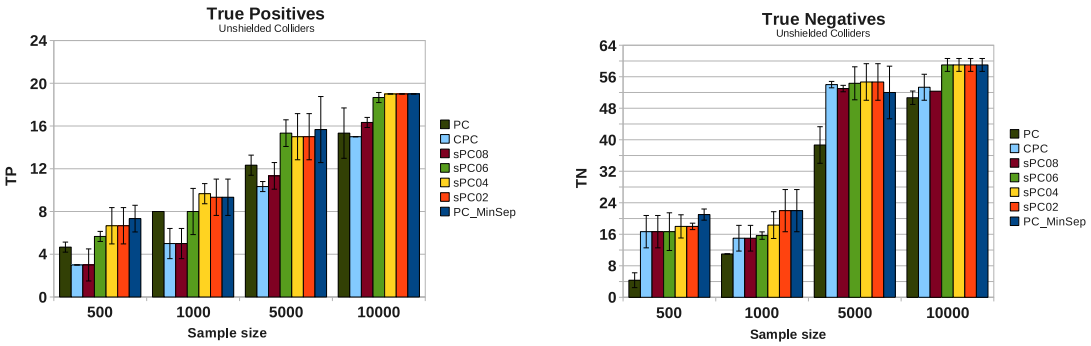


Figure 4.2: True positives and negatives on **unshielded colliders** (vs. unshielded triples) by various algorithms on the ALARM network.

Detecting unshielded colliders is a crucial step in causal discovery upon which the success of future steps rely. Figure 4.2 shows the true positives (TP) and true negatives (TN) obtained in detecting unshielded colliders on the ALARM network. The maximum value of the Y-axis on these charts denotes the true number of motifs in the ALARM network which has 24 unshielded colliders, and 64 unshielded triples which are not colliders. Across all small sample sizes we can see that almost all versions of the $sCPC_{or}$ algorithms and $PC_{minSepSet}$ outperform PC and CPC_{or}, by around 5, suggesting that the ALARM network has around 5 unshielded colliders that consistently obfuscate PC and CPC_{or} due to faulty CI tests that result in adjacency unfaithfulness. Note that while the performance of $sCPC_{or}$(0.8) is better than CPC_{or} it is quite close to it as well. $sCPC_{or}$(0.2) with a lower β better tolerates adjacency unfaithfulness and performs significantly better. $PC_{minSepSet}$'s performance is also quite good and comparable to the low-β $sCPC_{or}$

algorithms.

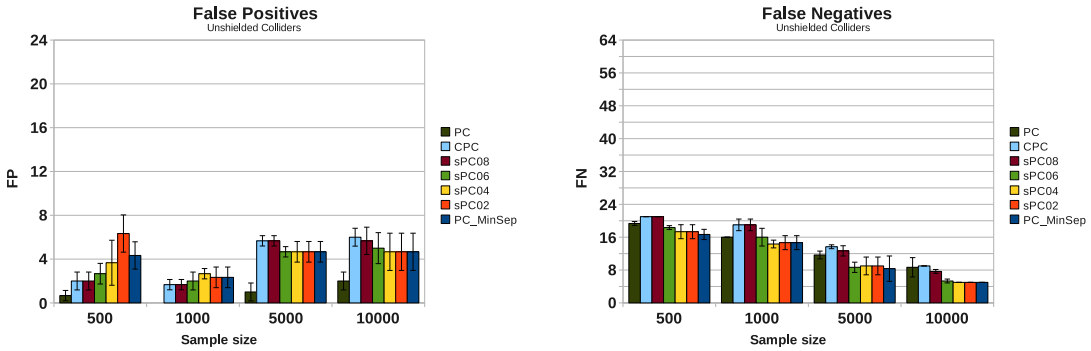


Figure 4.3: False positives and negatives on **unshielded colliders** (vs. unshielded triples) of various algorithms on the ALARM network.

The next pair of charts in Figure 4.3 shows the corresponding number of false positives (Type I errors) and false negatives (Type II errors) on the same tests as described above. With large enough sample size (≥ 5000) , $sCPC_{or}$ and $PC_{minSepSet}$ outperform CPC_{or} but PC performs best by detecting the least number of false positives. Since the PC encounters and stores the least number of separating sets for each removed edge, (just one), it detects the fewest number of unshielded colliders overall, explaining its low false positive rate.

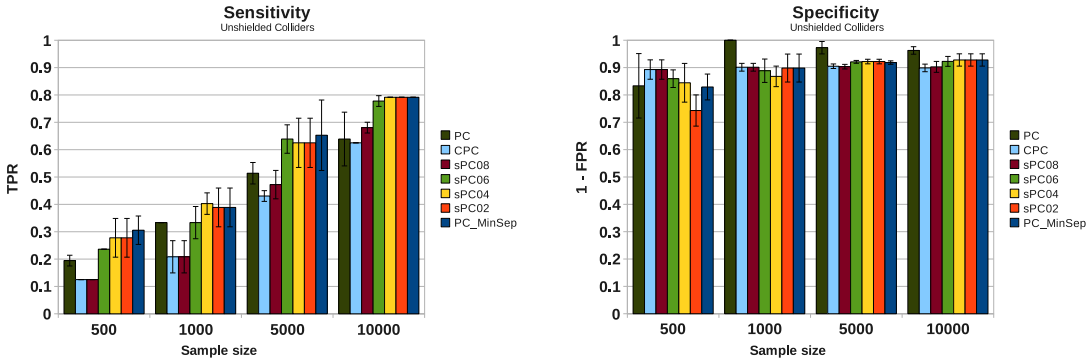


Figure 4.4: Sensitivity and Specificity on **unshielded colliders** (vs. unshielded triples) of various algorithms on the ALARM network.

Sample size →	500	1000	5000	10000
$PC_{minSepSet}$ vs. PC	6.32	2.48	3.26	4.92
$PC_{minSepSet}$ vs. CPC	10.99	6.2	5.39	52.44
$sCPC_{or}$ (0.2) vs. PC	3.59	2.48	3.58	4.92
$sCPC_{or}$ (0.2) vs. CPC	35.78	7.07	5.93	10.83

Table 4.1: Independent two-sample t_{test} statistics.

I summarize the results of figures 4.2 and 4.3 as sensitivity and specificity [1] metrics in figure 4.4. As a reminder, sensitivity (recall rate) measures the proportion of actual positives that are correctly identified, while specificity measures the proportion of negatives that are correctly identified.

$$Sensitivity = TPR = \frac{TP}{TP + FN}, \qquad Specificity = 1 - FPR = \frac{TN}{TN + FP} \quad (4.1)$$

where TPR stands for true positive rate and FPR stands for false positive rate.

Statistical significance : To test whether the improvements in performance for low-β $sCPC_{or}$ and $PC_{minSepSet}$ are statistically significant, I computed the independent two-sample t_{test} statistic, between $PC_{minSepSet}$ and $sCPC_{or}$(0.2), vs. PC and CPC [Box87]. The number of degrees of freedom for $n = 10$ trials is $d = 2n-2 = 18$. The corresponding single-tailed p-value of the t_{test} statistic is 1.734. To reject the null hypothesis that the difference in performance of these algorithms is due to chance, all the t_{test} statistics, should be above this threshold. Table 4.2 shows the computed t_{test} statistic for all sample sizes in the above experiment, establishing that the results are statistically significant.

From figure 4.4, and Table 4.2, we confirm that overall, low β $sCPC_{or}$ as well as $PC_{minSepSet}$ are clearly among the better of choices of algorithms for detecting unshielded colliders.

[1] Not to be confused with Yager's concept of specificity for fuzzy sets and possibility distributions [Yag08].

In the next pair of charts, (figure 4.5) we present the recalculated values of sensitivity and specificity after applying the deterministic orientation rules of the PC algorithm. Once again, we find that low-β $sCPC_{or}$ and $PC_{minSepSet}$ outperform both PC and CPC, and for the largest sample sizes tested, they are able to detect 82% of the true DAG edges while PC and CPC_{or} hover around 70%.

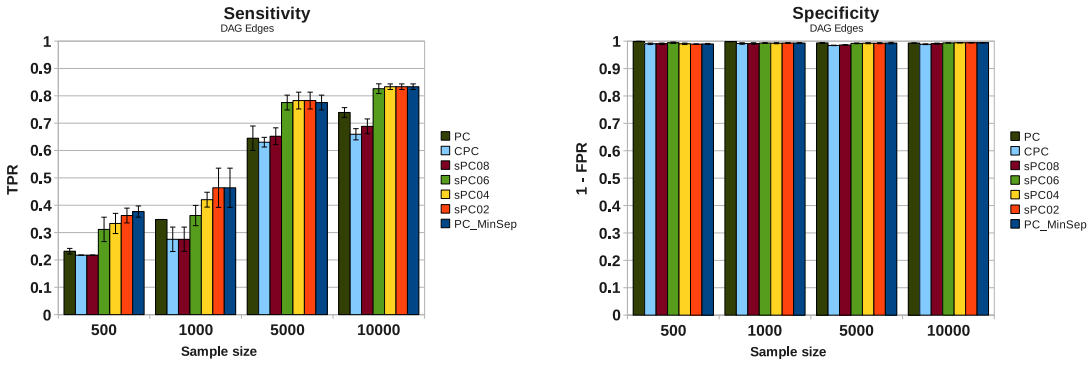


Figure 4.5: Sensitivity and Specificity on **DAG edges** of various algorithms on the ALARM network.

It is important to note that the success of finding unshielded colliders and consequently the DAG edges, depends primarily on the success of finding the undirected skeleton first. While the $sCPC_{or}$ and CPC_{or} come in to play only at the orientation phase, the PC_{sk} and $PC_{minSepSet}$ work at the edge removal phase. Figure 4.6 shows the relative performance of these algorithms and that $PC_{minSepSet}$ performs better on low sample sizes. However, with a large sample sizes (≥ 50000), both algorithms detect the skeletons perfectly. We do not show the corresponding specificity metric here as both algorithms have near-perfect specificity throughout all sample sizes.

Finally, I compare the running time of each algorithm with respect to the size of the dataset in figure 4.7. In this chart, the x-axis is a linear scale. As expected, the conservative algorithms $sCPC_{or}$ and CPC_{or} perform a much larger number of conditional independence tests making them the slowest. PC performs the fewest number of conditional independence tests and is fastest, while $PC_{minSepSet}$ performs a few more than PC

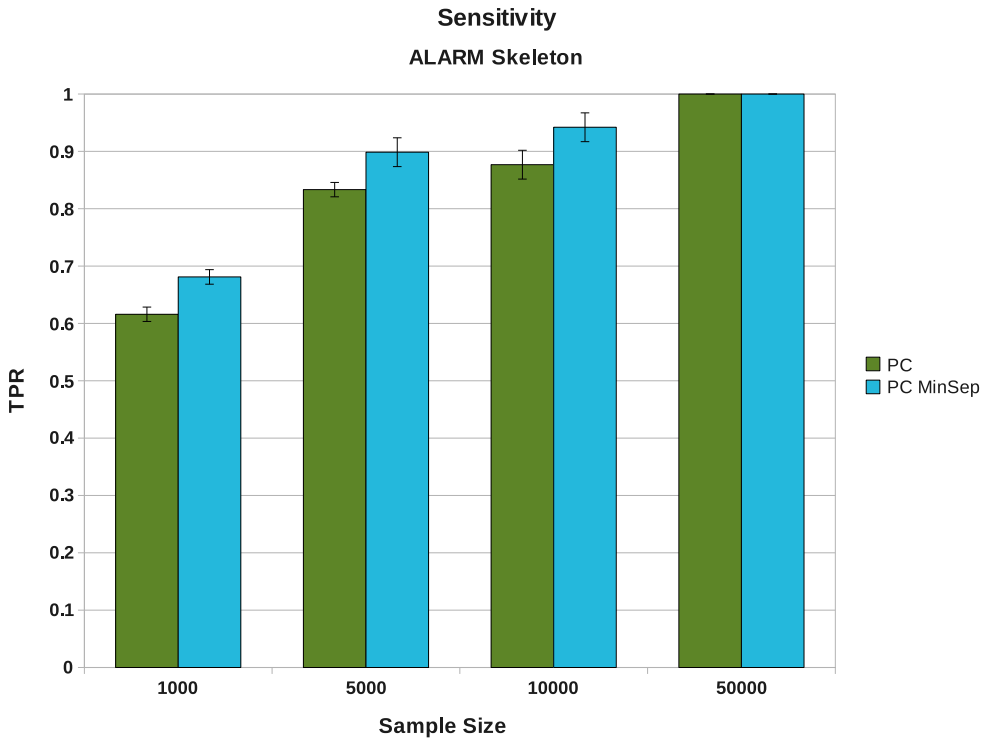


Figure 4.6: Sensitivity of PC vs. PC min Sep on finding the skeleton of the ALARM network.

but not as many as the conservative algorithms. The most important point to note is that $PC_{minSepSet}$ and PC are not significantly affected by sample size and therefore show the best promise for scalability.

From these results on the ALARM network, and especially by taking into account running time considerations, $PC_{minSepSet}$ is the winner. While there does seem to be utility in exercising some conservativeness by computing a larger number of relevant separation sets, the results indicate that the appropriate level of conservativeness is achieved by computing a few extra separation sets at the skeleton finding stage itself and choosing the separation set that entails the least conditional mutual information.

Next, I tested whether the significance level used in the χ^2 test for conditional inde-

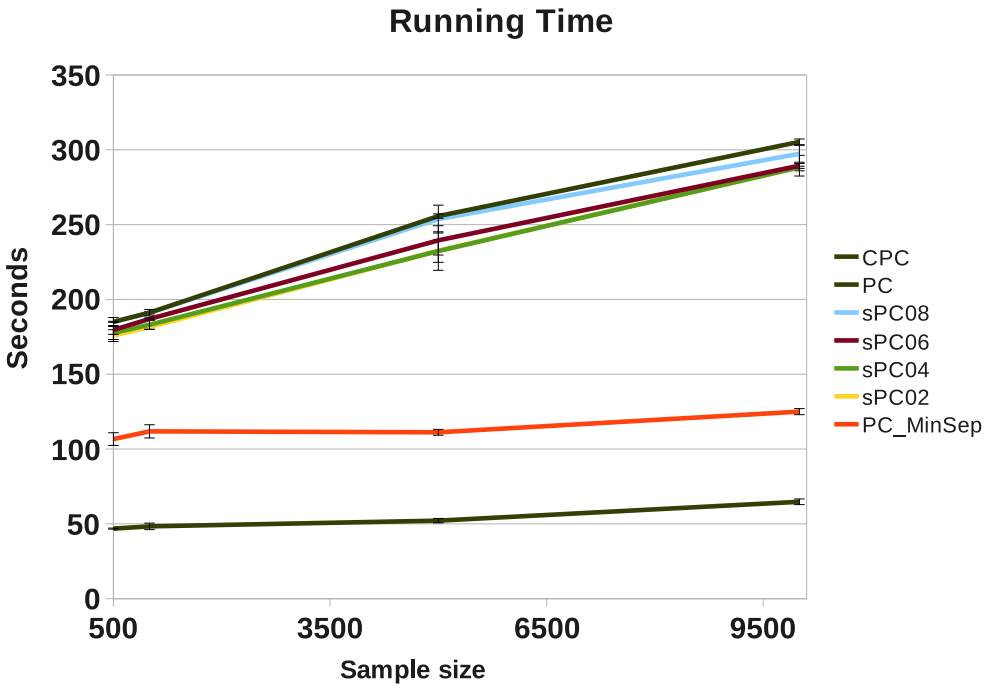


Figure 4.7: Running time of various algorithms on the ALARM network.

pendence had any effect on the performance of these algorithms. We tested three different significance levels (0.1, 0.05, and 0.01). The typical value chosen by statistical practitioners is 0.05, mainly due to historic reasons (it was suggested by Pearson [Pea04]) and also for preserving the uniformity of meaning of the term "statistically significant" across all scientific literature. However, it has also been suggested that the 0.05 value may not be suitable in certain domains in which case, a lower or greater value may be chosen with justification [J.71]. In the results as shown in figure 4.8, we do not see any consistent difference among the different significance levels on any of the algorithms, across all sample sizes. It is good to note that the χ^2 test is robust across a reasonably wide range of significance levels on the ALARM network and that the typical value of 0.05 can be reliably used for this domain and for learning causal Bayesian networks in general.

Next, I evaluate these algorithms on causal graphs of varying size (in terms of number

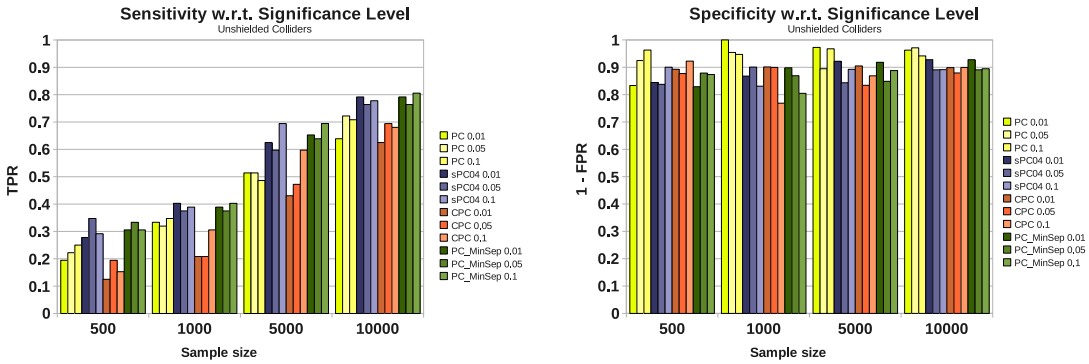


Figure 4.8: Effect of the significance level (χ^2_α) of CI testing on the Sensitivity and Specificity of various algorithms on the ALARM network.

of nodes). I generate random causal graphs between sizes 10 and 80 by the method described in 4.1.2. While doing so, I keep the other parameters of the graph similar to those of the ALARM network. Thus, all graphs I generate for this experiment have an average degree of $\frac{46}{37} = 1.24$ and a maximum degree of 6. As before, all results are presented as an average of 10 trials.

Figure 4.9 shows the performance of PC and $PC_{minSepSet}$ on finding the skeleton on these random graphs. A note on reading this chart: blue columns represent $PC_{minSepSet}$ while yellow columns represents PC. In the set of columns for each sample size, network size grows from left to right (10, 20, 30, 50, 80). and the corresponding comparable columns are placed next to each other (refer to legend). As seen before, both algorithms attain very good performance for large sample sizes. As expected, the algorithms perform better on smaller networks than large networks. However, $PC_{minSepSet}$ does not perform significantly better than PC in detecting the skeleton, in fact it is almost identical. From this, we can infer that even though the $PC_{minSepSet}$ provides no improvement in terms of the adjacency errors, the additional separation sets it computes is responsible for its better performance in the orientation phase.

While the two skeleton finding algorithms perform similarly on random graphs, there is

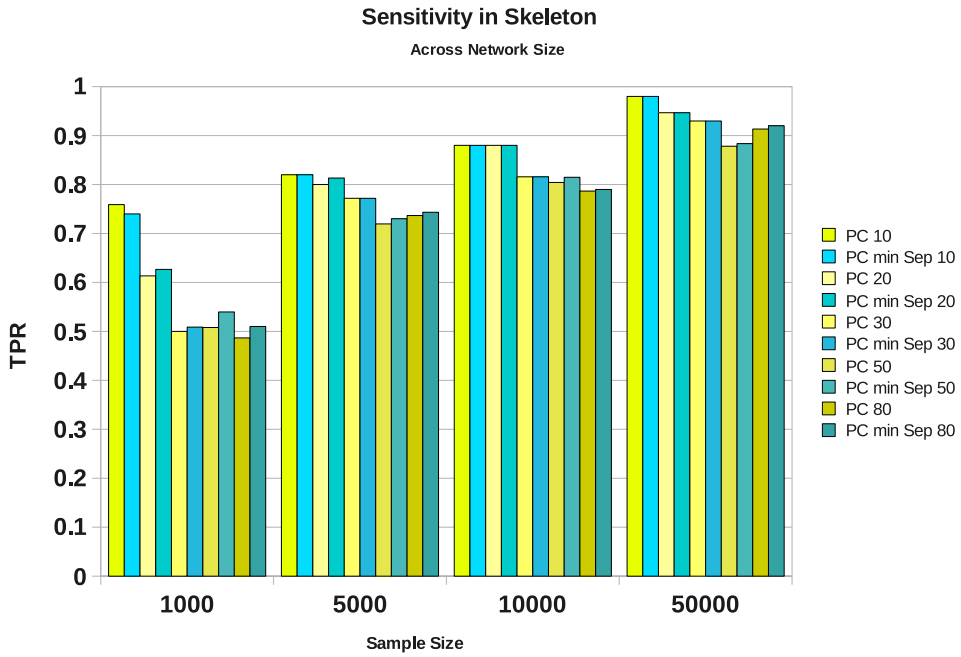


Figure 4.9: Sensitivity of finding the skeleton (undirected graph) across network and sample size.

a significant variation in the performance of the orientation algorithms. Figure 4.10 shows the sensitivity and specificity metrics of each algorithm {PC, $PC_{minSepSet}$, $sCPC_{or}$(0.2) & CPC_{or}} across network and sample sizes. A note on reading the chart: for each sample size, the first half of the columns (yellow and blue pairs) are the performances of PC and $PC_{minSepSet}$ across increasing network size and the right half of the columns (magenta & orange pairs) are the corresponding performances of low-β $sCPC_{or}$(0.2) and CPC_{or} The integer numbers on the legend denote network size.

PC and $PC_{minSepSet}$ have comparable performances everywhere, but are significantly and consistently less specific than both the conservative algorithms. $sCPC_{or}$ and CPC_{or} are however comparably specific. With respect to sensitivity, the low-β $sCPC_{or}$ is always better than any of the other algorithms.

We see that the overall sensitivities on random graphs is significantly less than the

sensitivities on the ALARM network (figure 4.4). This can be understood if one looks at the significant difference in the nature of the parameters of the ALARM network as opposed to the parameters assigned to these randomly generated models. As provided by the Norsys Software Corporation, and as commonly used in the ML community, the ALARM network's parameters render it nearly deterministic with a large majority of the CPT tables having one of their probabilities close to unity e.g. [0.99, 0.01]. On the other hand, the parameters of the randomly generated models are picked uniformly from the unit interval, which results in weaker correlations among the causal links than in the ALARM network.

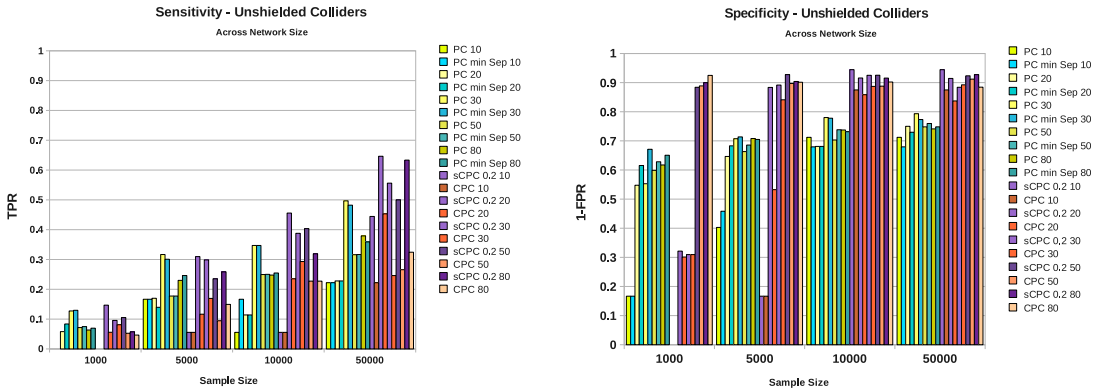


Figure 4.10: Sensitivity and Specificity in finding unshielded colliders across network and sample size.

Finally, I present the running time of these algorithms with respect to network and sample size. A theoretical analysis of the running time of the PC-family of algorithms is hard. It is known that PC is bounded by a polynomial of degree equal to the maximum degree of the nodes of the graph but the actual running time depends on the structure of the graph. The conservative algorithms also have the same upper bound on complexity, but tend to perform close to this upper bound, as they evaluate all possible conditional independence tests. We show empirical values for these running times (in seconds) in figure 4.11. All like-colored columns represent the performance of the same algorithm. For each set of columns in the same sample size, left to right denotes increasing network size.

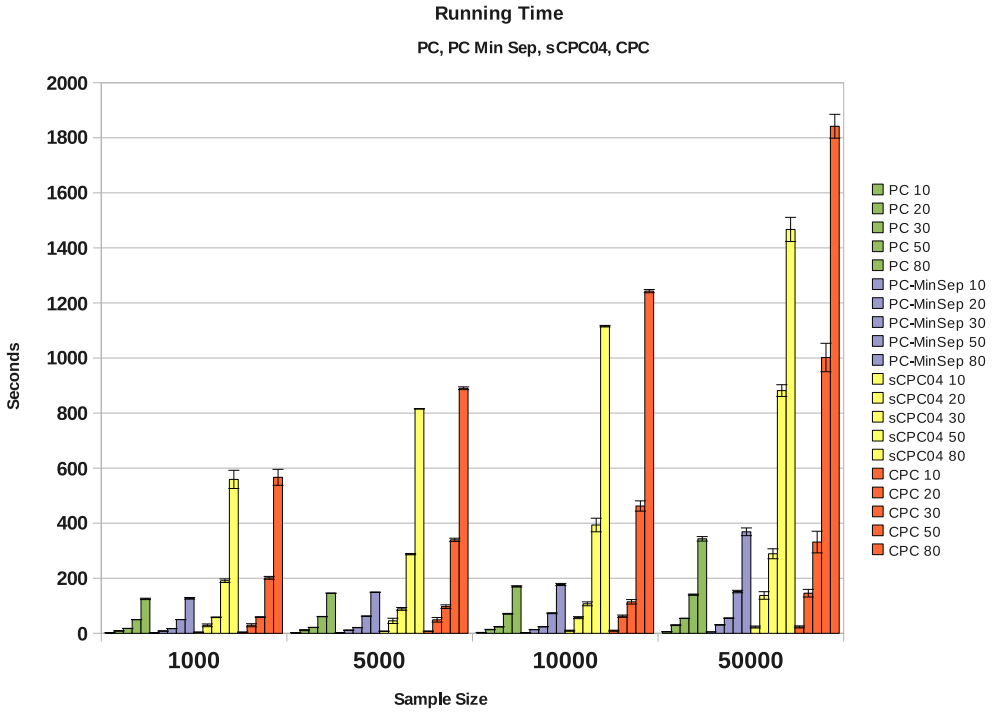


Figure 4.11: Running time across network and sample size.

As expected, the conservative algorithms are significantly more expensive than PC and $PC_{minSepSet}$, both of which have comparable running times. The additional conditional independence tests that $PC_{minSepSet}$ performs compared to PC does not seem to affect the running time significantly (green vs. blue columns). On the other hand, among the conservative algorithms $sCPC_{or}$ is consistently faster than CPC_{or}.

In conclusion, with the confidence obtained from the additional results on multiple graphs of varying size, I have the same result as stated for the ALARM network. The most reasonable algorithm to choose for practical use whenever we expect a certain number of violations of the faithfulness assumption in the causal graph along with the problem of imperfect conditional independence tests evaluated from finite datasets is $PC_{minSepSet}$. It evaluates a fewer number of conditional independence tests than the conservative algo-

rithms but still performs as well as the soft-conservative algorithm $sCPC_{or}$ in most cases. However, if computing time is not a concern and the problem domain is of a sufficiently tractable size, $sCPC_{or}$ should be used as it is more sensitive in the case of causal connections of weak strength.

4.3 Performance of Parent Search

In section 3.5 I described the *parentalSearch* algorithm to find the parents of the set of context variables when subjected to a hard intervention. In this section, we show its performance on the ALARM network. First we show the accuracy of the method on a single context setting. In our experiments testing this algorithm, we assume that none of the edges have been oriented and thus search for parent sets among all possible subsets of the neighbors for each node. Note that in the realistic case, the problem is less challenging: fewer number of neighbors will have to be taken into consideration on an average, as unshielded colliders may already be oriented. The algorithm is assessed as successful only when it finds the exact parent set and even if the detected parent set is off by one variable, I denote it as an error.

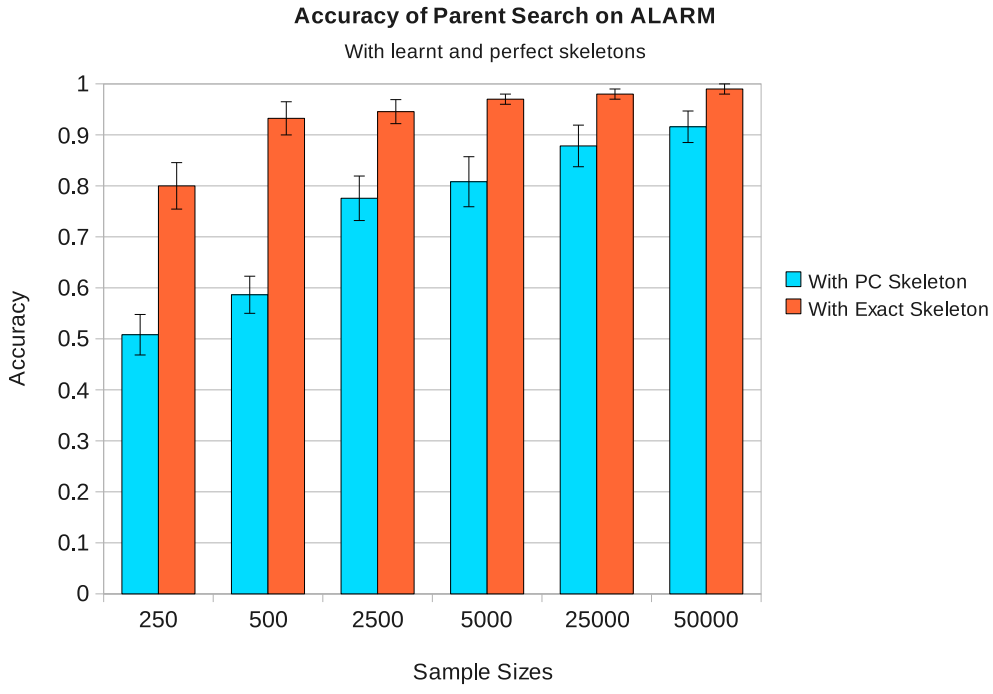


Figure 4.12: Performance of single variable context Parent Search.

Figure 4.12 shows the average accuracy obtained over a single context setting of each variable in the causal graph. Additionally, for each variable each result is an average obtained over 10 different sampled datasets of the denoted size. The error bars represent standard deviation. For single variable contexts, the *parentalSearch* performs quite well when provided with the exact skeleton, climbing very quickly to 95% accuracy even for small sample sizes and asymptotically reaches perfect performance with growing sample size. However, when the skeleton has errors (is computed by PC), its accuracy drops. This is understandable, as the *parentalSearch* algorithm searches in the space of the subsets of its neighboring nodes. If provided with the wrong set of neighbors, it is not searching the correct search space. The performance drop is thus, due to the PC and not due to *parentalSearch*.

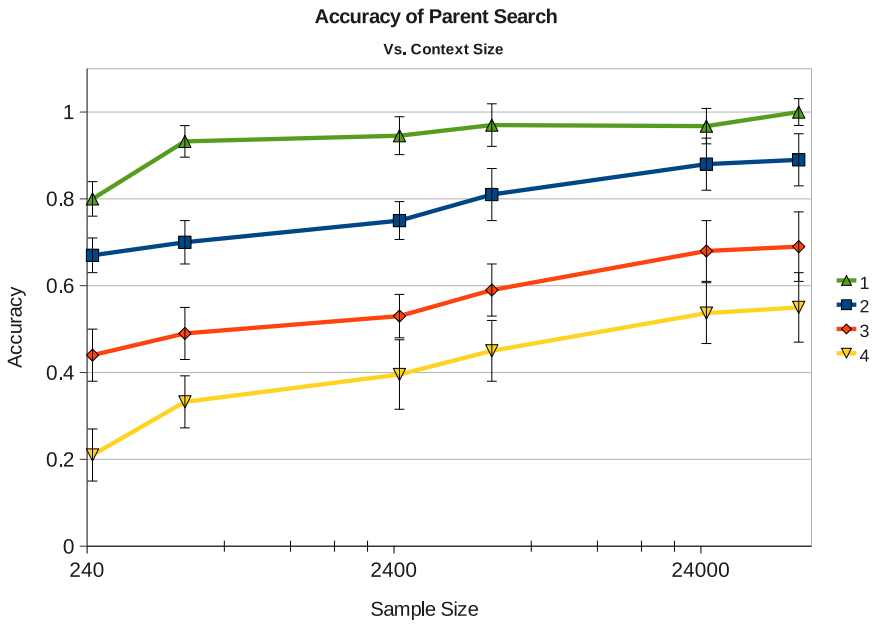


Figure 4.13: Performance of Parent Search vs. context size

Next, I test this algorithm in multiple variable contexts. For this experiment, I chose a set of $|C|$ variables at random, set them each at specific values. In Figure 4.13 I present the performance of *parentalSearch* on contexts of size {1,2,3 & 4}. Each point is an average

of 50 different randomly chosen contexts. The error bars represent standard deviation as usual. I already observed that the algorithm performs well for contexts of size 1, however there is significant drops in performance as context size increases. Accuracy is as low as 60% for contexts of size 4 even for very large sample sizes. This is due to the explosion of the search space. Each variable has a search space exponential in the size of its neighbors, which by itself is not too bad when I consider graphs of a manageable maximum degree. For multiple variable contexts, the search space becomes the Cartesian product of these spaces, and thus it quickly becomes very large. Thus, *parentalSearch* makes significantly more errors.

Additionally, I tested the performance of the parent search algorithm using the true KL-divergence, making the assumption that the actual distributions are known. This was performed on a Bayesian network with 15 variables, as true KL-divergence is practically infeasible beyond this limit. In this case, the parent sets for all sizes of contexts tested, one through four, were returned perfectly. When the correct parent set was used, the true KL-divergence metric was either *zero* or a very small floating point number of the order 10^{-23}, typical of inaccuracies in floating-point operations. Every other parent set returned with significantly higher values, several orders of magnitude greater. This reinforces the idea that theoretically, KL-divergence is a suitable metric to use for this application. In cases where the empirical KL-divergence metric is used for parent search, the probability of error involved with each hard intervention due to limited sample size, might translate to the wrong set of parents for a small percentage of interventions. If this is not taken into account, this might potentially magnify the error rates of future experiments. To address this, one might use the idea of redundancy as a method of verification and decrease the probability of error in orientation. For example, if an experiment on A, detects B as a parent of A, we can verify the link $B \rightarrow A$ by an experiment on B. However, this method will potentially require twice the number of experiments, but could be very valuable in domains with very weak causal links that are hard to detect from a single direction only.

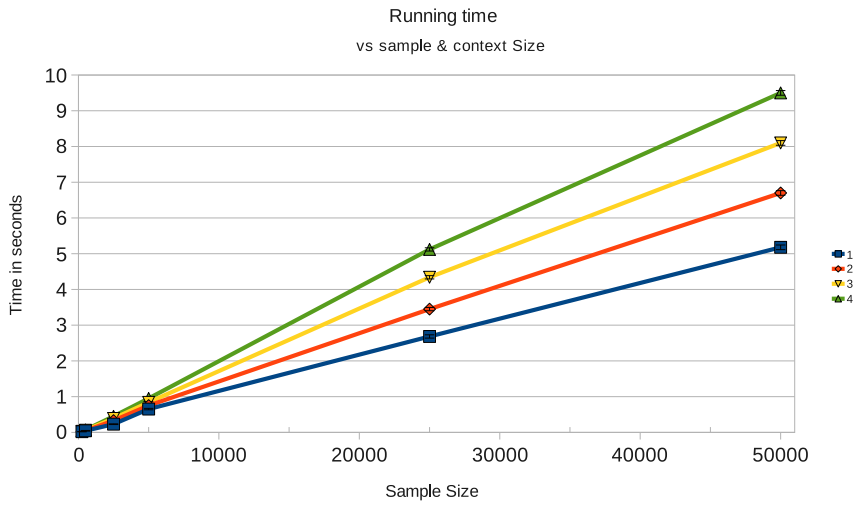


Figure 4.14: Running time of vs. context size and sample size

Figure 4.14 shows the corresponding running times across sample sizes. As a result, I think the *parentalSearch* can be used confidently for finding the parents of single variable hard interventions even when we have relatively low sample sizes. However, for contexts of larger size, the algorithm loses accuracy very quickly and should not be used. Nevertheless, the method is novel and improvements to its performance may be possible if a solution is found for the problem of the exploding search space for larger contexts.

4.4 Performance of approximate KL-Divergence

The *parentalSearch* algorithm required a fast algorithm to measure divergence between two distributions as estimated from their respective sampled datasets. I proposed and developed the *approximateKLD* for that purpose. However, I noted in section 3.5.1 that this algorithm can also be used in the general case for other purposes, especially for high dimensional distributions. Therefore, it is valuable to assess the performance of this approximate KL-Divergence measure on its own merit. I present these results in this section.

In experiments not reported here, I found that the closeness of the *approximateKLD* to the true KL-Divergence depends on the parameters of the Dirichlet prior adjustments used in the algorithm and it is very difficult (and I conjecture, impossible) to find a suitable setting for these parameters that is robust across a family of distributions. However, it was only necessary to find if the algorithm has good relative performance. In other words, given three distributions P_0, P_1 and P_2, *approximateKLD* maintains the same ordering as the true KL-Divergence, i.e. if $KLD(P_0, P_1) < KLD(P_0, P_2)$ then *approximateKLD*(P_0,P_1) < *approximateKLD*(P_0,P_2) and vice versa. For the purpose of testing this, we performed the following experiment on different networks of size 5, 10, 15 and 18. 18 was the maximum network size for which we can compute the exact KLD in a reasonably short time. First we randomly generate a causal Bayesian network of given size and set its parameters. This network represents P_0. Then, to generate P_1 and P_2, we first chose a random number of nodes ranging between 1 and 4, perform a perturbation on each of the selected nodes and compute whether the true KLD order matches the approximate KLD order. Figure 4.15 shows the average results of 1000 such runs across graph and sample size. Note that the algorithm performs very well even for small sample sizes of around 450 regardless of the graph size. However, the improvement in performance over sample size is rather slow.

To conclude this section, I can state that when a research task only requires a test for

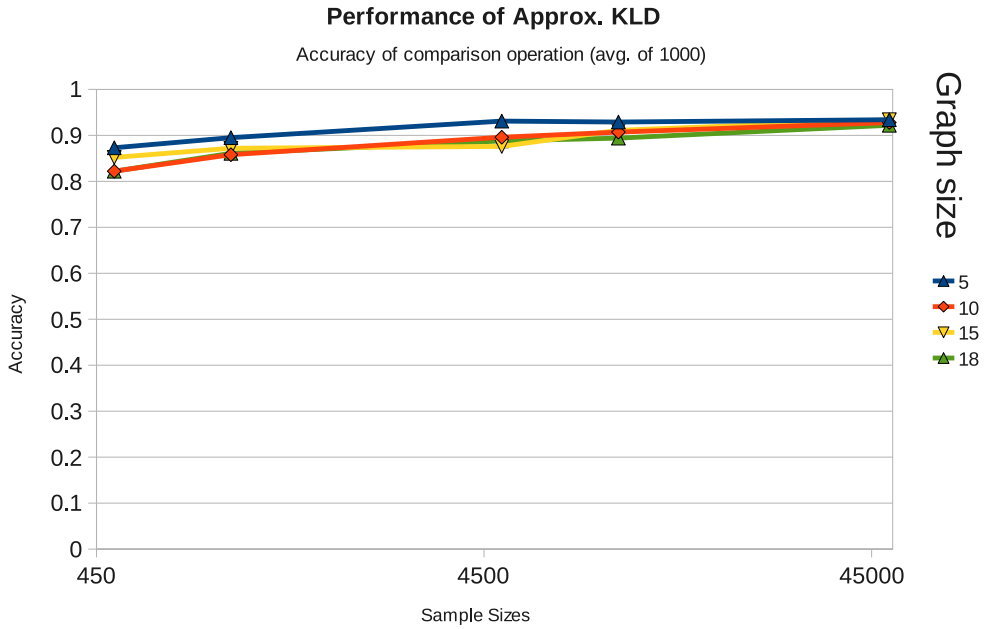


Figure 4.15: Performance of approximate KLD algorithm. Each point represents the mean of 1000 comparison tests for randomly chosen parametric changes on a causal Bayesian network.

relative divergence between distributions, the approximate KL-Divergence algorithm from data is useful in two scenarios.

1. When the true distribution is not available and only samples are available.

2. For distributions of large dimensionality, even when the true distribution is available, the true KLD computation will take $O(2^n)$, where n is the dimensionality of the distribution, whereas *approximateKLD* runs in $O(m \log m)$, where m is the size of the dataset.

4.5 Evaluation of Interactive Causal Discovery

In section 3.6, I hypothesized that a large class of causal models that are of interest to practitioners may be very different from the worst case causal graphs analyzed in earlier work. Such causal graphs do not require computationally expensive algorithms for determining experimental priority towards the goal of keeping the number of experiments low enough for efficient causal recovery. I proposed a simple and fast heuristic based on undirected degree, to compute the scores that translate into node priorities and described an interactive algorithm that uses this heuristic.

In this section, to evaluate the algorithm, I remove the role of the human experimenter in the algorithm and instead allow the causal learning agent to proceed as though a single experiment was performed on the node with highest priority at each step. For each causal model, we then evaluate the number experiments for every edge in the graph to be oriented. Most causal models of interest are relatively sparse, when compared to the worst case analyses of Eberhardt and others [ES06, MM06]. For example, the ALARM network, commonly used by the AI community as a benchmark, has only 2 cliques of size 3 and an average degree of 1.24. We are interested in finding out how many experiments are needed on the class of causal models similar to ALARM. Note that for ALARM, if one ignores the 2- and 3-cliques, the node scores based on vertices belonging to the maximum number of the largest cliques (now only cliques of size 2), simply boils down to the suggested heuristic, highest undirected degree.

I generate a large sample of graphs of varying network size, from 10 through 50, and for each graph size I set the average degree of the causal graphs to be ($\hat{d} = \frac{|E|}{|V|} = |V|^{K-1}$) for different values of K. I call K, the density index, and $1 \leq K < 2$. Note that for $K = 1$, $|E| = |V|$. Graphs with $|K| = 1$ are guaranteed not to be connected and likely to consist of several sparse disconnected subgraphs. On the other hand, when K is close to 2, the graphs approach fully connectedness, which have a high incidence of large cliques.

I vary K in steps of 0.1 from 1.0 to 1.7 and for each value of K and for each graph size, I generate 10,000 different causal models. Computing the mean number of experiments N_{exp}towards full causal discovery, gives the result as shown in figure 4.16.

The average number of experiments is initially high for sparse graphs, ranging from 2 to 8 for graph sizes of 10 to 50. For low K, the graphs are barely connected and are probably several disjoint subgraphs or have low values of separating cuts. Therefore, an orientation discovery in one subgraph does not help determine any new orientations in an of the other disconnected subgraphs, forcing us to carry out more interventions. As the connectivity of these graphs improves with K, $1.4 < K < 1.5$, a minimum of around 1.5 is reached in the average number of experiments, N_{exp}. With $K > 1.5$ larger cliques become more likely and the number of experiments required starts to rise again. With my method for random graph generation, I am unable to generate graphs with $K > 1.7$ efficiently, and therefore restrict our results within this range. However, intuition related to connectivity and earlier work by Eberhardt suggests that N_{exp} will keep increasing with K for all graphs of $K > 1.7$. Note that ALARM network would belong to the class of graphs as indicated by the pointer in figure 4.16.

The graphs generated for the above experiment belong to the class of *random graphs*. There are no high level structural constraints on the types of graphs generated other than the restriction based on average degree determined by the values of K. Graphs pertaining to specific domains might be constrained by a vast range of local and global properties. For example, there is a great deal of difference between "typical" causal models of biology, like gene regulatory networks, and causal models in industry, such as a factory control system. Biological causal models may have properties that are seemingly more random, and are constrained by the mechanisms of macro-molecular biology. Industrial causal models, being human designed tend to be more constrained and have a more intuitive causal flow. In such cases, special treatment and analysis of characteristics of these causal models are requred. In fact, an investigation into a domain specific characterization of causal models

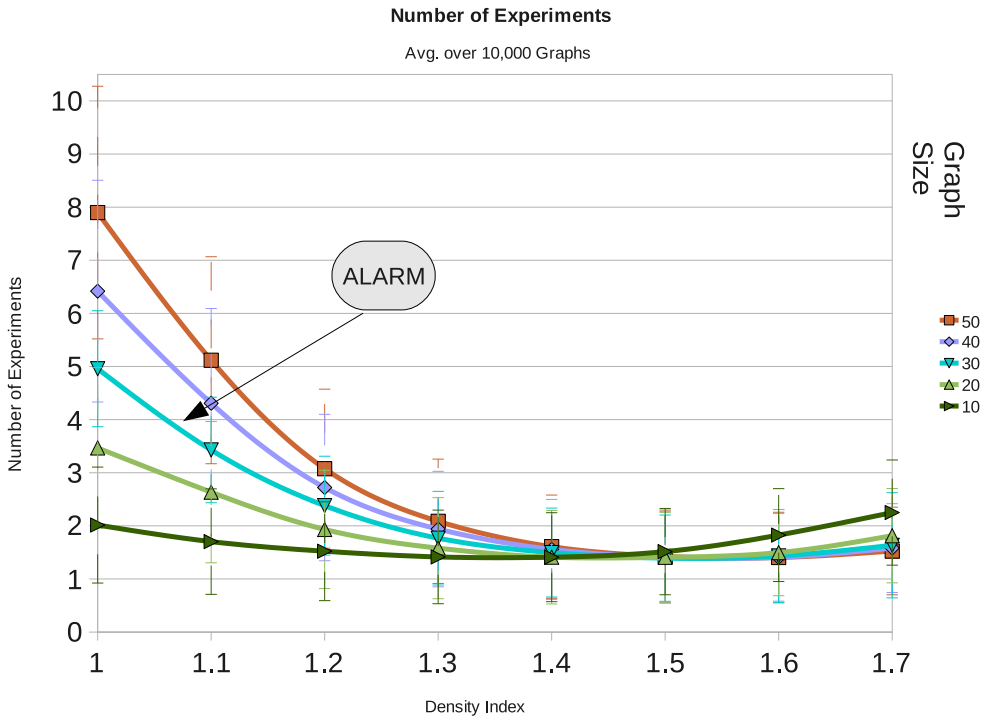


Figure 4.16: Number of Experiments required for full causal discovery over random graphs across different sizes and densities

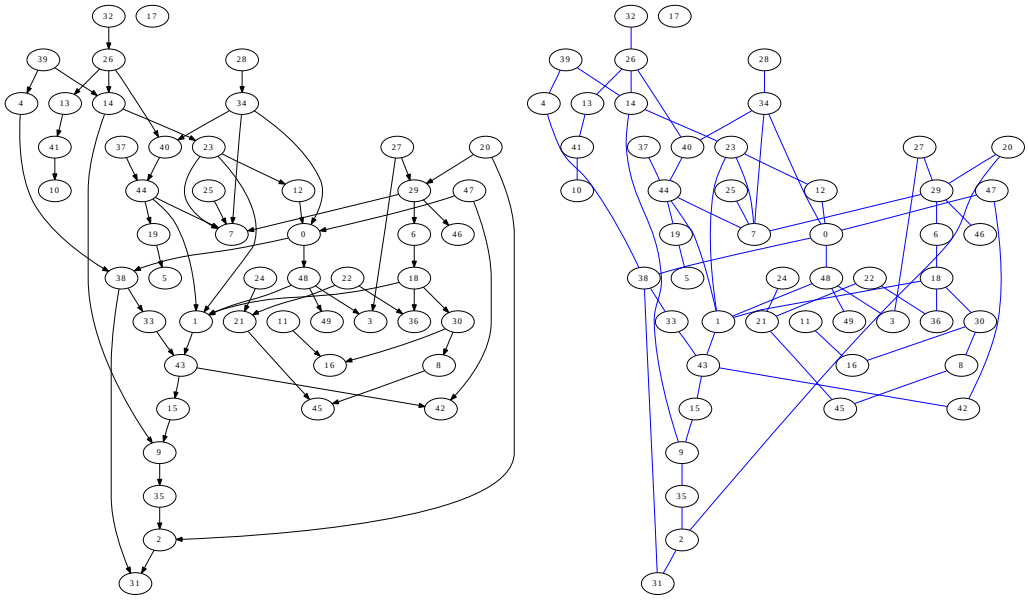
is suggested as an area of future research. The results of the above experiments should be interpreted as pertaining to causal models that have characteristics similar to the class of random graphs generated according to the description in Section 4.1.2.

In conclusion, while I may not be using the optimal order for experiments, these results suggest and reinforce the intuition that very sparse graphs also require a larger number of experiments close to number for the worst case of fully connected graphs. However, for a large class of graphs with with $1.3 \leq K \leq 1.7$ and with network size up to 80, large cliques do not dominate, and even a sub-optimal heuristic for finding node priorities can result in a very low number of experiments ($N_{exp} < 3$) towards complete orientation. Particularly, an interesting result is that the least number of experiments for all the graph sizes tested seems to have a minimum for $K \approx 1.5$.

$$N_{exp} \approx 1.5 \text{ for } |E| \approx |V|\sqrt{|V|} \tag{4.2}$$

4.5.1 A Demonstration

We conclude this chapter with a demonstration of the steps in the process of causal discovery on a randomly generated causal model of 50 nodes. For the sake of the demonstration we assume that all tests involving conditional independence, parent search and finding orientations by soft intervention, give perfect results.



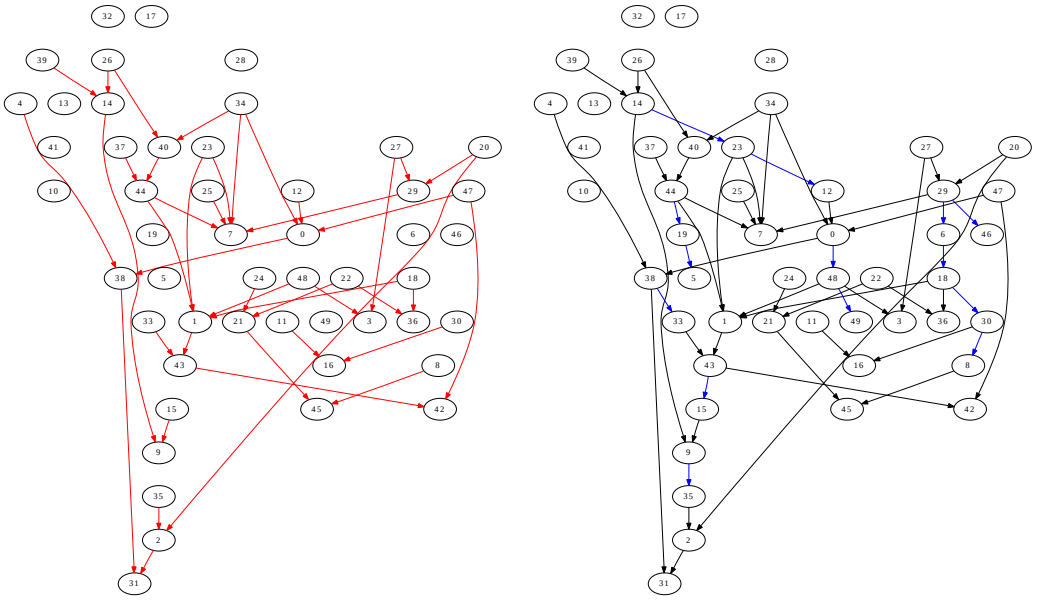
(a) The True Graph (b) Skeleton detected by PC

Figure 4.17: The first stage of causal discovery

Figure 4.17a shows the original true causal model. Based on observational data alone, the PC algorithm or its variants, finds the undirected skeleton shown in figure 4.17b.

The next phase in the PC algorithm finds all the unshielded colliders shown in figure 4.18a. This is followed by the application of the Meek rules. This signifies the limit of what can be learned by observational data alone on this causal model (figure 4.18).

Next, based on the node scores, the algorithm conducts the first experiment by a hard intervention on variable 32. *parentalSearch* detects that the set of parents of 32 is the



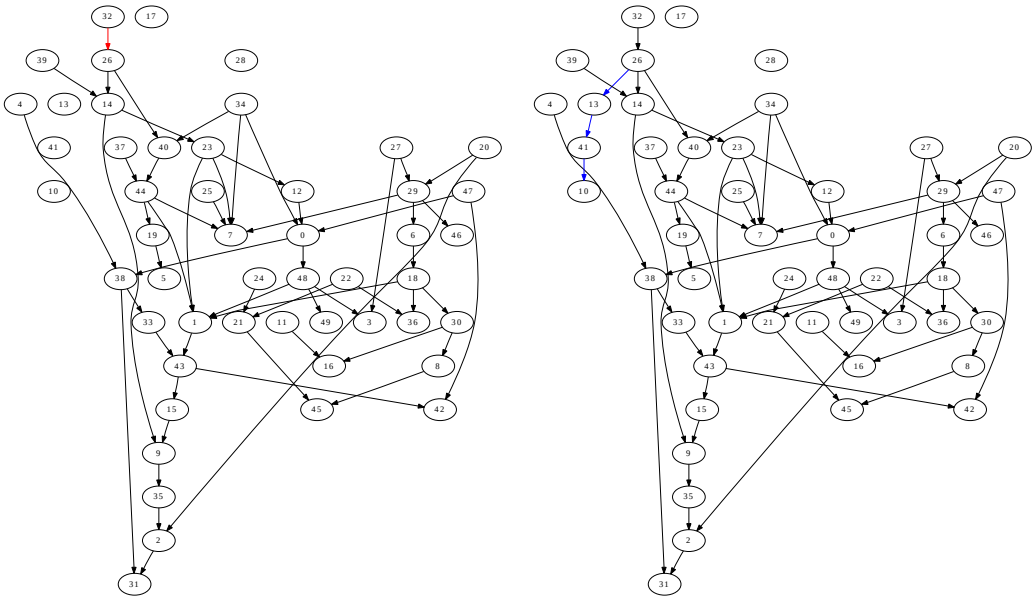
(a) Unshielded colliders detected

(b) First application of Meek orientation rules

Figure 4.18: The PDAG in 4.18b with 56 oriented edges and represents the limit of learning from observational data alone.

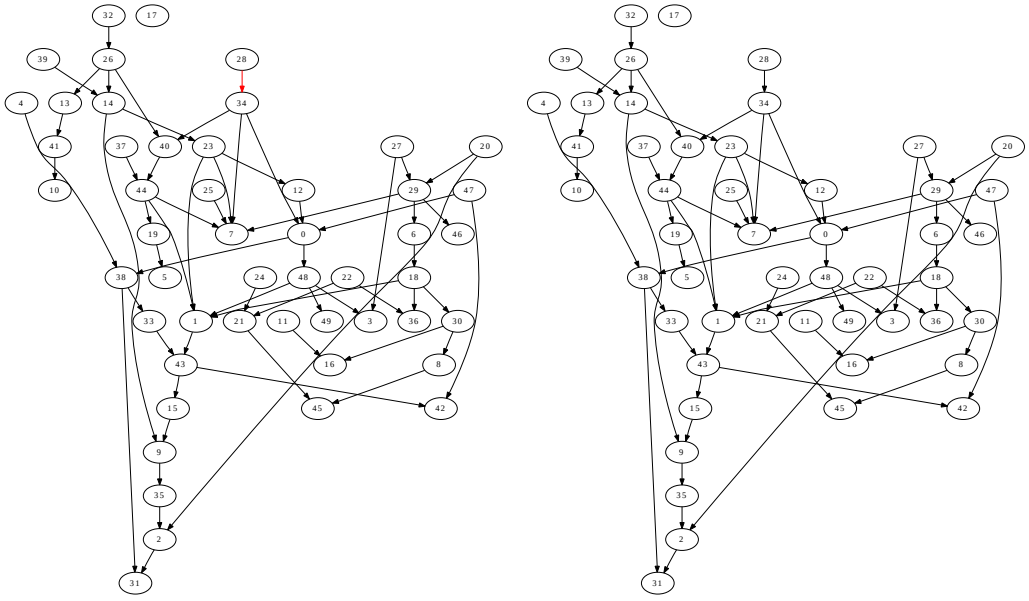
empty set, implying the orientation $32 \rightarrow 26$. Applying the meek orientation rules, we get the forced orientations $26 \rightarrow 13$, $13 \rightarrow 41$ and $41 \rightarrow 10$.

In the second experiment, a soft intervention on 28 is carried out, detecting the edge $28 \rightarrow 34$. No new orientations are found by the Meek rules. Finally, there remains only one edge to be oriented and it is determined as $39 \rightarrow 4$ through a soft intervention on node 39, resulting in the recovery of the full causal model.



(a) Hard intervention on node 32 (b) Meek orientation rules

Figure 4.19: Result after first experiment: 4 new orientations.



(a) Soft intervention on node 28 (b) Meek orientation rules

Figure 4.20: Result after second experiment: 1 new orientation.

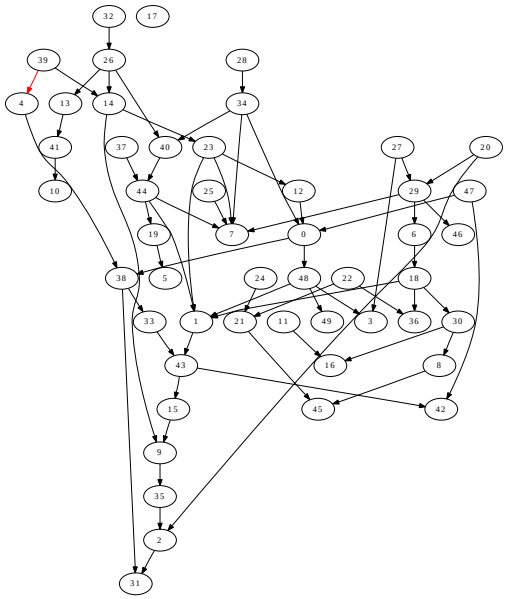


Figure 4.21: Soft intervention on node 39. Full causal graph recovered.

4.6 A Structural Equation Model

So far I have tested the performance of the causal learning algorithms described in Chapter 3 on data sampled from causal Bayesian networks. This section demonstrates that the algorithms can be applied just as successfully to a different causal modeling domain, namely a model described by a system of structural equations, as described in section 1.1. Recall that, Equation 1.2 describes a system of non-linear functional causal models while Equation 1.3 is a linearization of the same. In Section 1.4, I used an example pertaining to an aircraft monitoring system to motivate the discussion of an automated causal learning agent. The aircraft network of Figure 1.2 is modeled as a linear SEM with the following equations:

$$\begin{aligned} a &= r_a + u_a(\sigma_a) \\ s &= r_s + u_s(\sigma_s) \\ w &= r_w + u_w(\sigma_w) \\ g &= \alpha_a a + \alpha_s s + u_g(\sigma_g) \\ v &= \alpha_s s + \alpha_w w + u_v(\sigma_v) \end{aligned} \tag{4.3}$$

where a stands for cruise altitude, s for cruise speed, w for wind speed, g for gear vibration and v for wing vibration. The independent variables a, s and w vary along with the corresponding independent linear ramps r_x defined in the unit interval $(0, 1)$ with the additive Gaussian noise terms u_x. The noise terms u_x for each equation are independent of each other and are modeled as zero-mean Gaussians.

Five thousand continuous domain samples were generated from the above model. Each variables range in the continuous domain was partitioned into three equal ranges with threshold values at the one-third and two-thirds points to discretize the samples into the discrete domain $0, 1, 2$. This table of values was then used as input to the $PC_{minSepSet}$ algorithm. We calculated the performance of $PC_{minSepSet}$ on datasets generated in this

fashion with varying values of σ_x. The results of this experiment are presented as sensitivity and specificity metrics (average over 30 runs) in Figure 4.22.

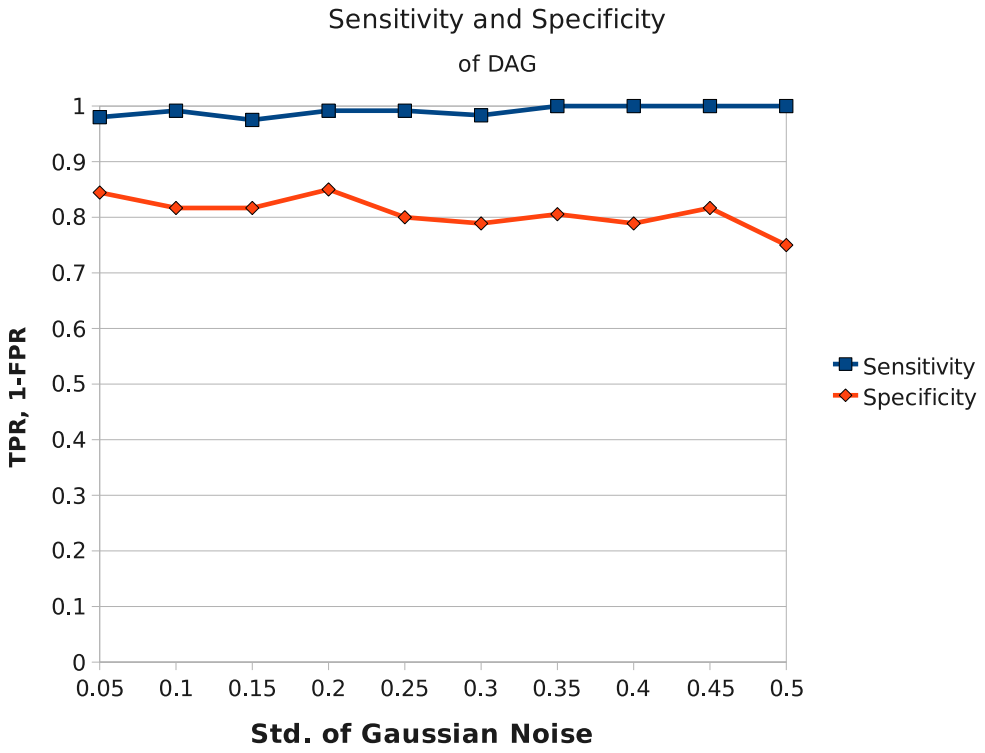


Figure 4.22: Sensitivity and Specifity of the $PC_{minSepSet}$ algorithm on a linear SEM model in finding the DAG of the Aircraft causal network.

Despite the simplistic discretization technique used in this example, note that the performance is very good and specificity only begins to drop for a very high level of noise. These results indicate that causal learning algorithms based on the Bayesian network formalism are very robust and can be adapted to other causal modeling domains with very little modification. Several areas in engineering that use approaches similar to SEM to model causal relationships can benefit from constraint based causal learning algorithms.

Chapter 5

Conclusion and Future Work

The primary motivation for this dissertation was the idea that "True AI" is possible only when algorithms are able to replicate what I believe is the most advanced cognitive ability of humans, namely, causal reasoning. Humans incessantly ask "why?", and have developed an ability and pioneered the discipline to carry out principled scientific investigation to determine the causal relationships of the world around then. AI and robots should be able to do that too. In essence, I am motivated by a future in which an AI system "understands", in the same quality that humans do.

Towards this goal, I investigated three different aspects of automated causal learning in this dissertation. In the next section, I summarize the main ideas of this research effort and draw some conclusions based on the results obtained. In section 5.2.1, I introduce how the concept of interaction information has a potential application in causal learning research. Finally, in section 5.2.2, I discuss the prospects of extending current formalisms and methods to temporal causal models and present some preliminary ideas.

5.1 Summary and Conclusion

My first research focus was on a family of algorithms that learn causal models from observational data alone. In sections 3.4 we presented the $sCPC_{or}$ and $PC_{minSepSet}$ algorithms as alternatives that traded off between the conventional greedy and conservative approaches in constraint based structure search algorithms. In section 4.2 we showed that these algorithms consistently achieve better performance than conventional approaches on several problem instances. I provided the justification and supported it with numerical results on finite-sample problem instances to show that they more robust to faulty conditional independence tests and violations of faithfulness.

Next, we explore the area of causal learning from experimental data, when experiments are performed as hard interventions. I introduced the *parentalSearch* algorithm in section 3.5 which infers the parental set of a node under a hard intervention. Corresponding results, presented in section 4.3, shows that this algorithm has very good accuracy for single variable contexts but suffers for larger contexts. These results suggest that the method is very successful when exploring the smaller single context variable search spaces, but is limited when exploring the Cartesian products of these search spaces. However, I claim that if we can find per-context variable parent sets independently, the algorithm can be scaled up for larger contexts. Therefore, incorporating better search strategies is a potential area for future research.

In developing the algorithm *parentalSearch*, I devised a new and computationally efficient method of approximating the Kullback Leibler divergence between two distributions from finite datasets. In section 3.5.1, I detail the algorithm *approximateKLD* and provide the results in 4.4. I argue the correctness of the algorithm and that it converges to the true KL divergence in the large sample limit. On relatively small datasets of high dimensional distributions, the approximation is not very reliable. However, the algorithm has a high degree of accuracy as a divergence comparator and I provide numerical results

to support this. Thus, in applications where one is interested only in such comparisons, and with distributions that are too large to compute the KL-divergence exactly (such as *parentalSearch*), this technique is an ideal solution.

Finally, I attack the problem of determining ideal sequences of experiments that reduce the number of experiments required for complete causal recovery. In section 3.6, I present the *interactiveCausalDiscovery* algorithm as a fast alternative to the conventional methods which are computationally expensive. While the conventional methods are claimed to work well on the worst cases, I show in section 4.5 that the fast method fares surprisingly well, for a large class of causal models, particularly for the class of models that are of great interest to the research community. From the results of the numerical simulation I show that causal graphs with average degree close to the square root of the number of vertices require the least number of experiments on average.

To conclude, the research implications of this dissertation are four-fold. I presented three new advancements to causal learning. Each is a solution to related but independent sub-problems in the field. The fourth contribution is a novel approximation algorithm for KL-divergence. All three advancements to causal learning are based on a a very general theoretical framework and are tested on domain independent problem instances, suggesting that they have a wide range of applicability to specific domains. Likewise, the KL-divergence approximation can not only be applied to the field of causal learning but also to the wider, more general area of probabilistic and stochastic modeling.

5.2 Future Research

During the course of the research that comprises this dissertation, several interesting questions and ideas arose. While they are beyond the scope of detailed discussion in the context of this dissertation, they are potential areas for future research and therefore demand some attention. At this point, I believe that three major research ideas dominate in terms of their

potential to improving current techniques and in extending the scope of causal learning research. In the next three sections, (5.2.1, 5.2.2 and 5.2.3) I briefly discuss these ideas, as fuel for further thought.

5.2.1 Using Interaction Information for Causal Learning

Interaction information [McG54, Bel03] is a generalization of mutual information. It expresses the amount of information shared by a set of variables, beyond the information that is present in any subset of these variables. Interaction information can be either positive or negative and this property has for a long time not been very well understood and perhaps has been the reason it has not been adopted widely as a measure of information by the research community. Recently, Jakulin and Bratko [JB03] developed a classification algorithm based on interaction information that relaxes the assumption that most conventional classifiers make; that the attributes are independent. They also discuss some visualization techniques based on interaction information [JB04].

In the three variable case, interaction information can be written as follows:

$$I(X;Y;Z) = I(X,Y|Z) - I(X,Y) \tag{5.1}$$

Consider the special case when $I(X,Y) > 0$ and $I(X,Y|Z) = 0$, corresponding to the case of negative interaction. This special case of negative interaction corresponds to the independence statement $(X \perp\!\!\!\perp Y|Z)$. Recall that the PC family of algorithms uses this information to mark the unshielded collider $X \rightarrow Z \leftarrow Y$. As a *gedankenexperiment*, consider a perturbation of the distribution on $\{X, Y, Z\}$ that renders $I(X,Y|Z) = \epsilon$, where ϵ is a small positive number, $I(X,Y)$ retains its value, and the negative interaction condition is still true. Given Z, X and Y become more dependent than when Z was not given.

- Does this imply a "shielded" collider at Z?
- Can interaction information be used to generalize the connection between conditional independence statements and *d*-separation?

If the answer to the above questions is "yes", it has great implications to causal learning research as well as Bayesian network formalism. There have been some prior attempts to generalize *d*-separation in the special case of belief propagation in polytrees [CD08], but a full generalization is yet unknown.

The difficulty arises from a confusing symmetric property of interaction information for the three variable case.

$$\begin{aligned} I(X;Y;Z) &= I(X,Y|Z) - I(X,Y) \\ &= I(X,Z|Y) - I(X,Z) \\ &= I(Z,Y|X) - I(Z,Y) \end{aligned} \tag{5.2}$$

Unfortunately, symmetry challenges our argument for the shielded collider at Z. One can speculate that the link with the lowest pairwise mutual information should be marked as the "shield", in the shielded collider. This means that knowledge of the third variable raises the information shared by the "shield" pair by a greater relative value than the other two pairs.

A 3-cliques is always a shielded collider due to acyclicity constraints, so identifying the position of the collider leaves only the orientation of the shield unresolved. Additionally, recall that the examples for "unfaithfulness" discussed in this dissertation also belong to the same class. These are wide ranging and important implications with a potential to resolve a large class of problems in probabilistic methods and I believe they provide sufficient motivation future research on interaction information, especially in the context of Bayesian networks.

5.2.2 Temporal Causal Models

It is a widely accepted fact that almost all causal relationships have a temporal delay between the cause and effect and in fact some of the earliest works on causality primarily rely on this fact [Gra69]. Research in neuroscience, econometrics and psychology heavily rely on learning causal relationships from time series data [Gra69, Che97]. Pearl points out in [Pea00] that temporal asymmetries and biases can be reconciled into Bayesian network framework using the concept of *statistical time*. He cautions however that temporal precedence alone is insufficient for causal inference due to our inability to measure quantities at the exact moment of occurrence. For example, we might note that the barometer dropped and soon after that, it rained. Barometers do not cause rain. In domains with little background information, it is easy to draw this erroneous conclusion if we used only temporal precedence for causal inference. Hence, modern research on causal learning research has focused primarily on modeling instantaneous based on the ideas of independence, counterfactuals and manipulability.

A popular extension to Bayesian networks to include the representation of temporal dependencies are *dynamic Bayesian networks* [Gha98]. Due to the acyclicity constraint, a limiting feature of Bayesian networks is in its inability to model cyclic causal relationships, which are a common occurrence in a large class of phenomenon. Dynamic Bayesian networks solve this problem elegantly by duplicating the instantaneous portion of the Bayesian network in two time slices, and introducing the "temporal" connections from nodes of the preceding time slice to the next. Dynamic Bayesian networks have been successfully used in several areas of machine learning and AI to model temporal causal relationships, the most popular of them being the *hidden Markov model* [Pea88, Gha02, Rab89]. Several techniques have been proposed for learning dynamic Bayesian networks from data and their success has been shown in a wide range of applications [Gha98, Mur02, SDW05, Pfe05, Zwe98].

However, the constraint based structure search community has not significantly attacked the problem of learning explicitly the temporal connections in causal models. It is an interesting area of future research, to investigate modifications to the PC-family of algorithms to learn from dynamic data streams. Key ideas about the correspondence between statistical and physical time detailed by Reichenbach and Pearl provide the basis and justification for such an effort [Pea00, Rei56].

5.2.3 Incorporating Background Knowledge

Throughout this dissertation we have considered the problem of learning causal structures from the point of zero causal knowledge. Except for the assumption of causal sufficiency and that we are aware of the variables in question, we make no assumption on any kind of prior knowledge about the causal network. In most cases of practical interest, the agent conducting the causal inquiry often has some facets of information about causation already. Typically, sources of background causal information are: expert knowledge, information from a previously conducted experiment in another domain, knowledge about physical constraints, etc. Such background information can be of several types. We enumerate some of them here.

1. There is information (presence or absence) about a direct causal link between two variables.

2. There is information about an adjacency between two variables.

3. There is information about a causal path between two variables, but not about the constituents of the path.

4. There is information about an abstract property about the causal graph, for example, one set of variables are connected to another set of variables only through a (small) third set (or $\emptyset$) of variables.

We can concieve of other kinds of information as well, but these already indicate the interesting array of possibilities that categorize the structures of background causal information. Additionally, each piece of background information may be available with a varying degree of uncertainty, depending on its source or its applicability to the domain in question. Some of these types of prior knowledge may naturally be incorporated into the Bayesian structure learning framework as priors, but it remains an interesting question about how such information can be incorporated into the constraint-based and intervention based causal learning algorithms. The PC family of algorithms make use of constraints in the form of conditional independence statements. A natural way to incorporate background information is by rewriting these other forms of information into statements about conditional independence and use them to bootstrap the PC algorithm. This motivates future research into choosing suitable representations for the different types of background information and finding effective methodologies to automatically translate these representations into conditional independence statements.

One can also expect conflicts between uncertain background information and information learned from data. In such cases, one needs to be able to quantify the uncertainty in both types of information and determine principled methodologies for conflict resolution. Currently there are no established measures to quantify the degree of uncertainty related to specific independence statements learned from data. Experience suggests that measures based on deviation from zero conditional mutual information and sample size can be devised. The problem of conflict resolution is furher complicated by 'long range' implications of a conflict. In some cases, the effects of a conflict might not remains local. A 'flip' in an independence statement might affect the structure of the causal graph several links away due to implications of *d*-separation and acyclicity.

Another approach would be to learn from data from scratch and manually edit the causal graph after learning. This approach might be conceptually simpler, but even so, quantifications of uncertainty about causal structure is necessary for the human editors to

make an unbiased and informed revision of the causal graph. The area of causal learning with facilities to incorporate several types of background knowledge is thus, a promising direction for future research.

References

[Alc05] Josep Roure Alcobé. Incremental methods for bayesian network structure learning. *AI Commun.*, 18(1):61–62, 2005.

[Aria] Aristotle. Metaphysics v 2.

[Arib] Aristotle. Physics ii 3.

[Aric] Aristotle. Posterior analytics, book 2, part 11.

[Bel03] Anthony J Bell. The co-information lattice. In *Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation: ICA 2003*, 2003.

[BG95] Hilary Buxton and Shaogang Gong. Advanced visual surveillance using bayesian networks. In *In International Conference on Computer Vision*, pages 111–123, 1995.

[Box87] Joan Fisher Box. Guinness, gosset, fisher, and small samples. *Statistical Science*, 2:45–52, February 1987.

[BP63] Thomas. Bayes and Richard Price. An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[BSCC89] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. In *Second European Conference on Artificial Intelligence in Medicine*, volume 38, pages 247–256, London, Great Britain, 1989. Springer-Verlag, Berlin.

[Bun91] W. L. Buntine. Theory refinement of Bayesian networks. In *Uncertainty in Artificial Intelligence*, 1991.

[Car83] Nancy Cartwright. *How the Laws of Physics Lie*. Oxford University Press, 1983.

[CD08] Arthur Choi and Adnan Darwiche. Many-pairs mutual information for adding structure to belief propagation approximations. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pages 1031–1036. AAAI Press, 2008.

[CH92] Greg F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[Che97] Patricia W. Cheng. From Covariation to Causation: A Causal Power Theory. *Psychological Review*, 104:367–405, April 1997.

[CL68] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

[Coo90] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks (research note). *Artif. Intell.*, 42(2-3):393–405, 1990.

[Cor] Norsys Software Corp. The Norsys Bayes Net Library. "http://www.norsys.com/networklibrary.html".

[CY99] Gregory F. Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In *In UAI*, pages 116–125. Morgan Kaufmann, 1999.

[Dav04] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[Doy90] Sir Arthur Conan Doyle. *The Sign of Four*. Lippincott's Monthly Magazine, 1890.

[Ebe06] Frederick Eberhardt. Sufficient condition for pooling data from different distributions. In *In First Symposium on Philosophy, History, and Methodology of Error*, 2006.

[Ebe08] Frederick Eberhardt. Almost optimal intervention sets for causal discovery. In *Proceedings of the Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 161–168, Corvallis, Oregon, 2008. UAI Press.

[Ebe10] Frederick Eberhardt. Causal Discovery as a Game. *Journal of Machine Learning Research*, 6:87–96, 2010.

[EGS05] Frederick Eberhardt, Clark Glymour, and Richard Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In *In Proceedings of the 21st Conference on Uncertainty and Artificial Intelligence*, pages 178–184, 2005.

[EGS06] Frederick Eberhardt, Clark Glymour, and Richard Scheines. *N-1 Experiments Suffice to Determine the Causal Relations Among N Variables*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 97–112. Springer, 2006.

[EM00] Daniel Eaton and Kevin Murphy. Belief net structure learning from uncertain interventions. *Journal of Machine Learning Research*, 1:1–48, 2000.

[EM07] D Eaton and K Murphy. Exact bayesian structure learning from uncertain interventions. In *AI and Statistics*. Press, 2007.

[ES06] Frederick Eberhardt and Richard Scheines. Interventions and Causal Inference. In *Philosophy of Science Assoc. 20th Biennial Mtg*, 2006.

[ES07] Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

[FG96] Nir Friedman and Moises Goldszmidt. Learning Bayesian networks with local structure. In *Uncertainty in Artificial Intelligence*, pages 252–262, 1996.

[Fis29] R. A. Fisher. Tests of significance in harmonic analysis. In *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, volume 125, pages 54–59. The Royal Society, 1929.

[FMR98] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *in UAI*, pages 139–147. Morgan Kaufmann, 1998.

[FW95] John Ferron and William Ware. Analyzing Single-Case Data: The Power of Randomization Tests. *The Journal of Experimental Education*, 63:167–178, 1995.

[GauCE] Aksapada Gautama. Nyāya sūtra, 2nd Century C.E.

[GGS+04] A. Gopnik, C. Glymour, D. M. Sobel, L. E. Schulz, T. Kushnir, and D. Danks. A theory of causal learning in children: Causal maps and Bayes nets. *Psychological Review*, 111(1):3–32, 2004.

[Gha98] Zoubin Ghahramani. Learning dynamic bayesian networks. In *Adaptive Processing of Sequences and Data Structures*, pages 168–197. Springer-Verlag, 1998.

[Gha02] Zoubin Ghahramani. *An introduction to hidden Markov models and Bayesian networks*, pages 9–42. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002.

[GP93] Dan Geiger and Judea Pearl. Logical and Algorithmic Properties of Conditional Independence and Graphical Models. *The Annals of Statistics*, 21:2001–2021, December 1993.

[Gra69] C W J Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, July 1969.

[GS03] Peter Godfrey-Smith. *Theory and Reality, An Introduction to the Philosophy of Science*. Science and Its Conceptual Foundations. University of Chicago Press, 2003.

[Haa43] Trygve Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11:1–12, 1943.

[Hay94] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994.

[Hec96] David Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, 1996.

[Hec99] David Heckerman. A tutorial on learning with bayesian networks. In M. I. Jordan, editor, *Learning in graphical models*, volume 156 of *Studies in Computational Intelligence*, chapter 3, pages 301–354. MIT Press, Berlin, Heidelberg, 1999.

[HGC95] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20:197–243, 1995.

[Hol86] Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

[HP01] J. Halpern and J. Pearl. Causes and Explanations: A Structural-Model Approach — Part 1: Causes. In *Proc. of 17th Conf. on Uncertainty in AI (UAI-01)*, pages 194–202, 2001.

[Hum40] David Hume. A treatise of human nature, 1740.

[J.71] Bross I. D. J. Critical Levels, Statistical Language and Scientific Inference. In Godambe V.P. and Sprott, editors, *Foundations of Statistical Inference*. Holt, Rinehart and Winston of Canada, Ltd, 1971.

[JB03] Aleks Jakulin and Ivan Bratko. Analyzing attribute dependencies. In *PKDD 2003, volume 2838 of LNAI*, pages 229–240. Springer-Verlag, 2003.

[JB04] Aleks Jakulin and Ivan Bratko. Quantifying and visualizing attribute interactions: An approach based on entropy. *http://arxiv.org/abs/cs.AI/0308002 v3*, 308002:3, 2004.

[KB07] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

[KBDG04] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 180–191. VLDB Endowment, 2004.

[KC01] T. Kocka and R. Castelo. Improved learning of bayesian networks, 2001.

[KD05] Stanley Kok and Pedro Domingos. Learning the structure of markov logic networks. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 441–448, New York, NY, USA, 2005. ACM.

[KL51] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KRO+09] Ross D. King, Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N. Soldatova, Andrew Sparkes, Kenneth E. Whelan, and Amanda Clare. The Automation of Science. *Science*, 324(5923):85–89, 2009.

[KS04] M. Koivisto and K. Sood. Exact bayesian structure discovery in bayesian networks, 2004.

[LB94] Wai Lam and Fahiem Bacchus. Learning bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(4):269–293, 1994.

[Lew73] David Lewis. Causation. *Journal of Philosophy*, 70:556–67, 1973.

[Lew86] David Lewis. On the plurality of worlds, 1986.

[LJY07] Wei Li, Xiaoming Jin, and Xiaojun Ye. Detecting change in data stream: Using sampling technique. In *ICNC '07: Proceedings of the Third International Conference on Natural Computation*, pages 130–134, Washington, DC, USA, 2007. IEEE Computer Society.

[McG54] William McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, 1954.

[Mee95] Christopher Meek. Causal inference and causal explanation with background knowledge. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 403–441, San Francisco, CA, USA, August 1995. Morgan Kaufmann.

[Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[MM06] Stijn Meganck and Bernard M. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In *In Modeling Decisions in Artificial Intelligence, LNCS*, pages 58–69, 2006.

[Mur02] Kevin Patrick Murphy. Dynamic bayesian networks: Representation, inference and learning, 2002.

[NS76] Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: symbols and search. *Commun. ACM*, 19(3):113–126, 1976.

[Pea04] Karl Pearson. *On the Theory of Contingency and Its Relation to Association and Normal Correlation*. Drapers' Company, 1904.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Pea95] J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82:669–710, 1995.

[Pea00] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[Pfe05] Avi Pfeffer. Asynchronous dynamic bayesian networks. In *In Proc. UAI 2005*, 2005.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[RCKW05] Da Ruan, Guoqing Chen, Etienne E. Kerre, and Geert Wets, editors. *Intelligent Data Mining: Techniques and Applications*, volume 5 of *Studies in Computational Intelligence*. Springer, 2005.

[RD06] M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62(1–2):107–136, Feb 2006.

[Rei56] Hans Reichenbach. *The Direction of Time*. University of California Press, 1956.

[Ris78] J Rissanen. Modelling by the shortest data description. *Automatica*, 14:465–471, 1978.

[Rob76] R.D. Robinson. Counting unlabeled acyclic digraphs. In *Australian Conference on Combinational Mathematics*, pages 28–43, 1976.

[ROR07] K. Strimmer R. Opgen-Rhein. Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *Bioinformatics*, 8, Suppl 2:S3:21–54, 2007.

[Rub06] Donald B. Rubin. *Matched Sampling for Causal Effects*. Harvard University, Massachusetts, 2006.

[RZS06] Joseph Ramsey, Jiji Zhang, and Peter Spirtes. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 401–408, Arlington, Virginia, 2006. AUAI Press.

[SAB+10] Andrew Sparkes, Wayne Aubrey, Emma Byrne, Amanda Clare, Muhammed Khan, Maria Liakata, Magdalena Markham, Jem Rowland, Larisa Soldatova, Kenneth Whelan, Michael Young, and Ross King. Towards robot scientists for autonomous scientific discovery. *Automated Experimentation*, 2(1):1, 2010.

[SDW05] Sumit Sanghai, Pedro Domingos, and Daniel Weld. Relational dynamic

bayesian networks. *Journal of Artificial Intelligence Research*, 24:2005, 2005.

[SG09] Ricardo Silva and Zoubin Ghahramani. The hidden life of latent variables: Bayesian learning with mixed graph models. *J. Mach. Learn. Res.*, 10:1187–1238, 2009.

[SGS00] Peter Spirtes, Clark Glymour, and Richard Scheines. Constructing Bayesian Network Models of Gene Expression Networks from Microarray Data. In *In Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems & Technology*, 2000.

[SGS01] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, USA, second edition, January 2001.

[Sim53] Herbert Simon. Causal ordering and identifiability. *Studies in Econometric Method*, pages 49–74, 1953.

[SLS07] Nikita A. Sakhanenko, George F. Luger, and Carl R. Stern. Managing dynamic contexts using failure-driven stochastic models. In *FLAIRS Conference*, pages 466–471, 2007.

[SMR95] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *In Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 499–506. Morgan Kaufmann, 1995.

[Sow00] John F. Sowa. *Processes and Causality*, chapter 4, pages 206–245. Brooks Cole Publishing Co., 2000.

[SPB94] Wolfgang Spohn, Abteilung Philosophie, and D Bielefeld. On the properties

of conditional independence. In *Suppes: Scientific philosopher*, pages 173–194. Kluwer, 1994.

[SRLS08] Nikita A. Sakhanenko, Roshan Rammohan, George F. Luger, and Carl R. Stern. A new approach to model-based diagnosis using probabilistic logic. In *FLAIRS Conference*, pages 678–683, 2008.

[TKP06] Jin Tian, Changsung Kang, and Judea Pearl. A characterization of interventional distributions in semi-markovian causal models. In *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*, pages 1239–1244. AAAI Press, 2006.

[TP01a] J. Tian and J. Pearl. Causal discovery from changes. In *In: Proceedings of UAI 2001*, pages 512–521. Morgan Kaufmann, 2001.

[TP01b] J. Tian and J. Pearl. Causal discovery from changes: a bayesian approach. Technical report, In Proceedings of UAI 17, 2001.

[VP90] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *UAI '90: Proceedings of the Sixth Annualma Conference on Uncertainty in Artificial Intelligence*, pages 255–270, New York, NY, USA, 1990. Elsevier Science Inc.

[VP91] Thomas Verma and Judea Pearl. A theory of inferred causation. In *Second Internation Coference on the Principles of Knowledge Representation and Reasoning*, Cambridge, Massachusetts, April 1991.

[WB05] Dan Wu and Cory J. Butz. On the complexity of probabilistic inference in singly connected bayesian networks. In Dominik Slezak, Guoyin Wang, Marcin S. Szczuka, Ivo Düntsch, and Yiyu Yao, editors, *RSFDGrC (1)*, volume 3641 of *Lecture Notes in Computer Science*, pages 581–590. Springer, 2005.

[Web02] Andrew R. Webb. *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.

[Whe47] William Whewell. *History of the Inductive Sciences, from the earliest to the present time*. D. Appleton and Company, 1847.

[Wri21] Sewall S. Wright. Correlation and Causation. *Journal of Agricultural Research*, 20:557–585, 1921.

[Yag08] Ronald R. Yager. Measures of specificity over continuous spaces under similarity relations. *Fuzzy Sets Syst.*, 159(17):2193–2210, 2008.

[Zha08] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

[Zwe98] Geoffrey Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, 1998.