Advances in Causal Structure Learning

Theory and Algorithms

Constantin Aliferis, Gregory Cooper, Andre Elisseeff, Isabelle Guyon, and Peter Spirtes (Eds.)

Challenges in Machine Learning, Volume 8

2014

Foreword

Welcome to the revolution.

The essays in this volume represent recent (circa 2007) contributions to the extraordinary changes in our understanding of the possibilities for causal inference that began in the 1980s and that continue to advance. That revolution concerns both how to discover causal relations and how to predict with them, and papers in this collection address both aspects.

Twenty-five years ago a distinguished statistician, Terry Speed (who was in part responsible for the revolution, although he did not welcome it) remarked to me that regression may be an unreliable method to search observational data for causal relations, but no other method should be used. Notwithstanding, in recent years new methods in the family of procedures to which many of the present contributions belong have been applied in economics, epidemiology, molecular biology, mineralogy, neuropsychology, space physics, and many other domains, in several cases with experimental confirmation of learned structures, and in many other cases outperforming "conventional" statistical methods for forecasting and prediction. At about the same time, I asked another distinguished statistician, my colleague, Stephen Fienberg, if he had available any "large" datasets — meaning a dozen variables, give or take — that could be used for testing search algorithms. He told me there was no point in assembling such data sets because 12 variables was too big to allow analysis. Recent work has made experimentally confirmed causal inferences from observational data sets with tens of thousands of variables. The capacities for scientific discovery and prediction have changed radically, and they continue to.

The revolution is beginning to enter into advanced statistical textbooks and monographs, and into scientific practice, notably in neuroscience and genomics, where the discovery of causal relations in high dimensional data presents complexity problems that would have been insuperable anytime in the last century. Textbook traditions carry big inertia, but I hope the time is not far off when modern causal inference is taught to undergraduates as part and parcel of statistical estimation and prediction, and textbook slogans ("correlation is not causation" — true but misleading; "no causes in, no causes out" — false and misleading) are dispensed with. Meanwhile, the fundamental research exemplified in this volume marches on, with new, astonishing revelations tumbling over one another, year after year, from every direction on the planet, only the poles excepted. Heaps of important open problems remain. The challenges are as exciting as the accomplishments.

I have the guilty pleasure of disciplinary jingoism in noting the direct and indirect contributions of professionally trained philosophers to the work described here. Peter Spirtes' foundational work is well known, and Jiji Zhang, Thomas Richardson and Chris Meek are important contemporary investigators, but it is a pleasant surprise to find that work on defeasible reasoning by the late John Pollock has found an indirect causal path to the effort. Steven Weinberg, the Nobelist in physics, has written that at its best

philosophy of science is only "a pleasing gloss" on the history of science. Clearly, some introductions are needed.

Clark Glymour
Pittsburgh, July, 2014

Preface

Introduction

Learning valid causal mechanisms from non-experimental data and performing successful inference from such models about the expected effects of manipulating the model system (or even counterfactual inferences about interventions that *could* have happened but did not), has been for a long time considered an impossibility with an almost taboo status. "Correlation is not causation", the famous warning by R.A. Fisher, would often signal the end of the discussion with otherwise sophisticated and well-intentioned researchers.

As often happens in science, many insurmountable obstacles end up being overcome by ingenuity, inspiration, hard work, and even a little bit of luck. With regards to the latter, Nature often lends a helping hand by providing data generating functions and distributions that are relatively friendly to many types of discovery techniques, including causal ones.

Today we (computer scientists, philosophers of science, applied mathematicians, engineers, statisticians, econometricians, psychometricians, actuarialists, financial quantitative analysts, and all other "tribes" of data scientists) are well aware and have embraced a panoply of methods and practical tools that allow discovery of high quality qualitative causal models from data, and parameterization of those to derive quantitative causal models, and inference to predict the effects of manipulations of certain modelled variables. In short, not all correlation is causation but certain correlations *are* causal, and tools that allow us to make the distinction are now available, and indeed widely used.

How did we get here? And where do the contributions in the present volume stand between the foundations of successful non-experimental causal discovery analysis and practically deployed methods of today and of the future?

We can summarize the trajectory of discovery and innovation in this area and as it relates to these questions comprising the triptych:

⟨ Foundations, Translation to practice (from theory to practical algorithms and applications), New possibilities ⟩.

Foundations

With the benefit of decades of historical hindsight we can point to the work of the Nobel Laureate Herb Simon in the 1950s, of the Turing Award winner Judea Pearl (in the 1980s and onward) (a contributor to the present volume), of P. Spirtes (a contributor and co-editor of the present volume), Clark Glymour and Richard Scheines in the 1980s and onward, of G.F. Cooper (another co-editor of the volume) (1980s and onward), and of the Nobel Laureate Clive Granger (from the 1960s and onward) and their collaborators, students, and many others as truly essential. Together this community of multi-disciplinary researchers established a range of rigorous methods that under broad distributional assumptions can learn high quality causal models from observational data

or mixtures of observational and experimental data, (even identifying the presence of hidden variables that confound correlations in the data), overcoming the limitations that had been noted by R.A. Fisher regarding the perils of over-interpreting *all* correlations as causative.

The resulting innovations include important causal discovery algorithms and procedures — indicatively we mention the IC*, PC, FCI, K2, algorithms, the BDe and BDeu scoring metrics, the Granger test for causality in time series, the Do-Calculous (and the Front-Door and Back-Door criteria for conditioning and inference with causal models), and numerous extensions and refinements. Collectively they provide the basis for learning accurate causal models from data even in the absence of experiments, and for predicting the effects of manipulations on the data "in silico".

Peter Spirtes' contribution to the present volume ("Introduction to Causal Inference") provides a concise introduction to core ideas and methods about the Foundations and prepares the reader for new material in the present volume and beyond.

Translation to practice (from theory to practical algorithms and applications)

The contributions by Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos ("Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation" and "Part II: Analysis and Extensions") set out to describe an ambitious agenda of algorithmic and applied research at the bridge of the world of predictive modeling and of causal discovery. Their paper summarizes prior key work and provides generalized algorithmic frameworks for learning local and global causal models, and optimal predictor sets (Markov Blankets and Boundaries). The paper also compares representative algorithms from these frameworks to general machine learning and statistics methods, in terms of causal and predictive modeling performance across dozens of real and synthetic datasets.

Complementary to, and mutually reinforcing with the above effort, is the contribution of Jean-Philippe Pellet, André Elisseeff ("Using Markov Blankets for Causal Structure Learning") who also set out to answer causal questions using hybrid causal/predictive (Markov Blanket) techniques with promising practical results in terms of scalability and quality of discovery.

The paper by Raanan Yehezkel, Boaz Lerner ("Bayesian Network Structure Learning by Recursive Autonomy Identification") addresses the problem of learning high quality causal graphs efficiently via the novel recursive autonomy identification (RAI) algorithm. The algorithm learns causal structure by sequential application of conditional independence (CI) tests, edge direction and structure decomposition into autonomous sub-structures combining edge discovery and direction "from the outset and along the procedure". A large experimental evaluation shows strong results in the datasets used and in direct comparison against many of the top comparator algorithms.

Yang-Bo He, and Zhi Geng's paper ("Active Learning of Causal Networks with Intervention Experiments and Optimal Designs") deals with the very practical and experimental-science-relevant situation where causal induction has specified an equivalence class and experiments or quasi-experiments can then be deployed to refine the equivalence class. The paper proposes exact and approximate designs for sequential or batch experimentation to accomplish these goals.

New possibilities: pushing the frontier of what is feasible

The contribution by Tyler J. VanderWeele, and James M. Robins ("Properties of Monotonic Effects on Directed Acyclic Graphs") develops a number of probabilistic properties concerning monotonic effects and weak monotonic effects. These properties give rise to certain inequality constraints that provide new ways to test for the presence of hidden or unmeasured confounding variables that go beyond those already available in the literature.

The paper by Facundo Bromberg, and Dimitris Margaritis ("Improving the Reliability of Causal Discovery from Small Data Sets Using Argumentation") deals with a common weakness of modern constrain-based causal learning algorithms, that is the often low statistical power of employed conditional independence tests (CITs). The paper proposes enhancing the quality of CITs by using argumentation logics and introduces a new "argumentative independence test" with promising initial empirical results.

The contribution by Changsung Kang, Jin Tian ("Markov Properties for Linear Causal Models with Correlated Errors") introduces new theoretical results related to testing linear structural equation models with correlated errors. The results have direct implications for improving the state of the art in the identification of such models from data.

The contribution by Ilya Shpitser, and Judea Pearl ("Complete Identification Methods for the Causal Hierarchy") provides a framework for going from simpler (predict effects of prior interventions or natural occurrences) to intermediate (predict effects of possible interventions) to harder (counterfactually estimate effects of hypothesized past actions, different than the ones actually taken or observed). These layers provide a hierarchy of causal inference that maps cleanly to common questions in science and everyday life. Both theoretical and algorithmic results are provided to solve the identification problem for causal effects by providing a graphical characterization for non-identifiable effects, and algorithms for computing identifiable effects.

The paper by Jiji Zhang ("Causal Reasoning with Ancestral Graphs") introduces new mathematical results that extend prior results by Pear et al and Spirtes et al and open the door for causal intervention inference with partial ancestral graphs (PAGs). Given that in many practical situations the ability to learn causal graphs is confined within an equivalence class of the generative function and that PAGs can represent such classes, the work promises to address a vexing problem, that of inference within a data-consistent equivalence class approximation of the true causal generative function.

Conclusion

Perhaps, the causal discovery field resonates well today because "everything is possible" in the era of Big Data Science. Or perhaps it is the incredible success and potential of fields like causal discovery that makes data scientists feel that "everything is possible". Either way we hope that the readers will find the works presented in the present tome illuminating, inspiring and ultimately useful for their own work.

October 2014

The Editorial Team:

Constantin Aliferis
Center of Health Informatics and Bioinformatics
Department of Pathology, New York University
New York, NY 10016, USA
constantin.aliferis@nyumc.org

Gregory Cooper
Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA 15206
gfc@pitt.edu

André Elisseeff
Google Zürich,
Brandschenkestrasse 110,
8002 Zürich
elisseeff@google.com

Isabelle Guyon
ChaLearn
Berkeley, CA 94708, USA
isabelle@clopinet.com

Peter Spirtes
Department of Philosophy
Carnegie Mellon University
Pittsburgh, USA
ps7z@andrew.cmu.edu

Table of Contents

Foreword i

Preface iii

Introduction to Causal Inference 1
P. Spirtes; JMLR W&CP 11:1643–1662, 2010.

Using Markov Blankets for Causal Structure Learning 23
J.-P. Pellet & A. Elisseeff; JMLR W&CP 9:1295–1342, 2008.

Causal Reasoning with Ancestral Graphs 73
J. Zhang; JMLR W&CP 9:1437–1474, 2008.

Complete Identification Methods for the Causal Hierarchy 113
I. Shpitser & J. Pearl; JMLR W&CP 9:1941–1979, 2008.

Active Learning of Causal Networks with Intervention Experiments and Optimal Designs 155
Y.-B. He & Z. Geng; JMLR W&CP 9:2523–2547, 2008.

Markov Properties for Linear Causal Models with Correlated Errors 183
C. Kang & J. Tian; JMLR W&CP 10:41–70, 2009.

Improving the Reliability of Causal Discovery from Small Data Sets Using Argumentation 213
F. Bromberg & D. Margaritis; JMLR W&CP 10:301–340, 2009.

Properties of Monotonic Effects on Directed Acyclic Graphs 253
T.J. VanderWeele & J. M. Robins; JMLR W&CP 10:699–718, 2009.

Bayesian Network Structure Learning by Recursive Autonomy Identification 273
R. Yehezkel & B. Lerner; JMLR W&CP 10:1527–1570, 2009.

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation 319
C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani & X.D. Koutsoukos; JMLR W&CP 11:171–234, 2010.

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part II: Analysis and Extensions 387
C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani & X.D. Koutsoukos; JMLR W&CP 11:235–284, 2010.

JMLR: Workshop and Conference Proceedings 11:1643–1662, 2010 Submitted 2/10; Published 5/10

Introduction to Causal Inference

Peter Spirtes PS7Z@ANDREW.CMU.EDU
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Lawrence Saul

Abstract

The goal of many sciences is to understand the mechanisms by which variables came to take on the values they have (that is, to find a generative model), and to predict what the values of those variables would be if the naturally occurring mechanisms were subject to outside manipulations. The past 30 years has seen a number of conceptual developments that are partial solutions to the problem of causal inference from observational sample data or a mixture of observational sample and experimental data, particularly in the area of graphical causal modeling. However, in many domains, problems such as the large numbers of variables, small samples sizes, and possible presence of unmeasured causes, remain serious impediments to practical applications of these developments. The articles in the Special Topic on Causality address these and other problems in applying graphical causal modeling algorithms. This introduction to the Special Topic on Causality provides a brief introduction to graphical causal modeling, places the articles in a broader context, and describes the differences between causal inference and ordinary machine learning classification and prediction problems.
Keywords: Bayesian networks, causation, causal inference

1. Introduction

The goal of many sciences is to understand the mechanisms by which variables came to take on the values they have (that is, to find a generative model), and to predict what the values of those variables would be if the naturally occurring mechanisms were subject to outside manipulations. For example, a randomized experiment is one kind of manipulation that substitutes the outcome of a randomizing device to set the value of a variable (for example, whether or not a particular new medication is given to a patient who has agreed to participate in a drug trial) in place of the naturally occurring mechanism that determines the variable's value. In non-experimental settings, biologists gather data about the gene activation levels in normally functioning systems in order to understand which genes affect the activation levels of which other genes, and to predict what the effects of manipulating the system to turn some genes on or off would be. Epidemiologists gather data about dietary habits and life expectancy in the general population and seek to find what dietary factors affect life expectancy and predict the effects of advising people to change their diets. Finding answers to questions about the mechanisms by which variables come to take on values, or predicting the value of a variable after some other variable has been manipulated, is characteristic of causal inference. If only non-experimental data are available, predicting the effects of manipulations typically involves drawing samples from one probability density (in the

unmanipulated population) and making inferences about the values of a variable in a population that has a different probability density (in the manipulated population).

The rapid spread of interest in the last three decades in principled methods of search or estimation of causal relations has been driven in part by technological developments, especially the changing nature of modern data collection and storage techniques, and the increases in the processing power and storage capacities of computers. Statistics books from 30 years ago often presented examples with fewer than 10 variables, in domains where some background knowledge was plausible. In contrast, in new domains such as climate research (where satellite data now provide daily quantities of data unthinkable a few decades ago), fMRI brain imaging, and microarray measurements of gene expression, the number of variables can range into the tens of thousands, and there is often limited background knowledge to reduce the space of alternative causal hypotheses. Even when experimental interventions are possible, performing the many thousands of experiments that would be required to discover causal relationships between thousands or tens of thousands of variables is often not practical. In such domains, non-automated causal discovery techniques from sample data, or sample data together with a limited number of experiments, appears to be hopeless, while the availability of computers with increased processing power and storage capacity allow for the practical implementation of computationally intensive automated search algorithms over large search spaces.

The past 30 years has also seen a number of conceptual developments that are partial solutions to these causal inference problems, particularly in the area of graphical causal modeling. Sections 3 and 4 of this paper describe some of these developments: a variety of well defined mathematical objects to represent causal relations (for example, directed acyclic graphs); well defined connections between aspects of these objects and sample data (for example, the Causal Markov and Causal Faithfulness Assumptions); ways to compute those connections (for example, d-separation); and a theory of representation and calculation of the effects of manipulations (for example, by breaking edges in a graph); and search algorithms (for example, the PC algorithm). However, in many domains, problems such as the large numbers of variables, small samples sizes, and possible presence of unmeasured causes, remain serious impediments to practical applications of these developments.

The articles in the Special Topic on Causality (containing articles from 2007 to 2009) address these and other problems in making causal inferences. Although there are some superficial similarities between traditional supervised machine learning problems and causal inference (for example, both employ model search and feature selection, the kinds of models employed overlap, some model scores can be used for both purposes), these similarities can mask some very important differences between the two kinds of problems. This introduction to the Special Topic on Causality provides a brief introduction to graphical causal modeling, places the articles in a broader context, and describes the differences between causal inference and ordinary machine learning classification or prediction problems; it is not intended to provide a broad overview or a tutorial surveying all methods of causal inference.

Section 2 describes the problem of causal inference in more detail, and differentiates it from the typical machine learning supervised classification or prediction problem; Section 3 describes several different kinds of causal models; Section 4 describes some problems associated with search for causal models, and why algorithms appropriate for the discovery of good classification or prediction models in machine learning are not always appropriate for the discovery of good causal models; and Section 5 describes

some major open problems in the field. The various articles in the Special Topic on Causality are described throughout this article, depending upon which topic they address.

2. Manipulating Versus Conditioning

This section will describe three different kinds of problems (one typical machine learning or statistical problem, and two kinds of causal problems), and three different kinds of probability densities (conditional, manipulated, and counterfactual) that are useful for solving the problems.

2.1. Conditional Probabilities

Suppose that there is a population of individuals with the following random variables at time t: rw_t is the average number of glasses of red wine consumed per day in the 5 years prior to t, bmi_t is the body mass index of a person at time t, sex_t is the person's sex (0 = male, 1 = female) at time t, and ha_t is whether or not an individual had a heart attack in the 5 years prior to t. Since sex_t is rarely time-dependent, it will be replaced simply by sex.

Suppose an insurance company at time t wants to determine what rates to charge an individual for health insurance who has $rw_t = 1$, $bmi_t = 25$, and $sex = 0$, and that this rate is partly based on the probability of the individual having a heart attack in the next 5 years. This can be estimated by using the rate of heart attacks among the subpopulation matching the subject, that is $rw_t = 1$, $bmi_t = 25$, $sex = 0$. It is impossible to measure the values of ha_{t+5} at time t, because they haven't occurred yet, but if the probability density is stable across time, the density of ha_{t+5} among the subset of the population with $rw_t = 1$, $bmi_t = 25$, and $sex = 0$ will be the same as the density of ha_t among the subpopulation for which $rw_{t-5} = 1$, $bmi_{t-5} = 25$, and $sex = 0$. The density in a subpopulation is a conditional density, in this case $P(ha_t \mid rw_{t-5} = 1, bmi_{t-5} = 25, sex = 0)$.

Conditioning maps a given joint density, and a given subpopulation (typically specified by a set of values for random variables) into a new density. The conditional density is a function of the joint density over the random variables, and a set of values for a set of random variables.[1] The estimation of a conditional probability is often non-trivial because the number of people with $rw_{t-5} = 1$, $bmi_{t-5} = 25$, $sex = 0$ might be small. A large part of statistics and machine learning is devoted to estimating conditional probabilities from realistic sample sizes under a variety of assumptions.

If the insurance company is not attempting to change anyone's behavior then the question of whether drinking the right amount of red wine *prevents* heart attacks is irrelevant to their concerns; the only relevant question is whether the amount of red wine that someone drinks *predicts* heart attack rates. It is possible that people who drink an average of between 1 and 2 glasses of red wine per day for 5 years have lowered rates of heart attacks because of socio-economic factors that both cause average daily consumption of red wine and other life-style factors that prevent heart attacks. But even if moderate red wine consumption does not prevent heart attacks, the insurance company can still use the conditional probability to help determine the rates to charge.

If $\mathbf{X}$ is a set of measured variables, the conditional probability density $P(\mathbf{Y} \mid \mathbf{X})$ is not only useful for predicting future values of $\mathbf{Y}$, it is also useful for predicting current unmeasured values of $\mathbf{Y}$, and for classifying individuals in cases where $\mathbf{Y}$ is categorical.

1. In order to avoid technicalities, I will assume that the set of values conditioned on do not have measure 0.

2.2. Manipulated Probabilities

In contrast to the previous case, suppose that an epidemiologist is deciding whether or not to recommend providing very strong incentives for adults to drink an average of 1 to 2 glasses of red wine per day in order to prevent heart attacks. Suppose further that if adopted the incentives will be very widely effective. The density of heart attacks observationally *conditional* on drinking an average of 1 to 2 glasses of red wine per day is not the density relevant to answering this question, and the question of whether drinking red wine prevents heart attacks is crucial. Suppose drinking red wine does not prevent heart attacks, but the heart attack rate is lower among moderate red wine drinkers because some socio-economic variable causes both moderate red wine drinking and other healthy life-styles choices that prevent heart attacks. In that case, after the incentives to drink red wine are in place, the density of socioeconomic status among red wine drinkers will be different than prior to the incentives, and the conditional density of heart attacks among moderate red wine drinkers will not be the same after the incentives were adopted as prior to their adoption. Thus, using observational conditional densities to predict heart attacks after the incentives are in place will lead to incorrect predictions.

The density that is relevant to determining whether or not to recommend drinking a moderate amount of red wine is not the density of heart attacks among people who have chosen to drink red wine (choice being the mechanism for determining red wine consumption in the unmanipulated population), but the density of heart attacks among people who would drink red wine after the incentives are in place. If the incentives are very effective, the density of heart attacks among people who would drink red wine after the incentives are in place is approximately equal to the density of heart attacks among people who are assigned to drink moderate amounts of red wine in an experimental study.

The density of heart attacks among people who have been *assigned* to drink red wine (as opposed to those who have *chosen* to drink red wine, as is currently the case) is a *manipulated* density, that results from taking action on a given population - it may or may not be equal to any observational conditional density, depending upon what the causal relations between variables are. Manipulated probability densities are the appropriate probability densities to use when making predictions about the effects of taking actions ("manipulating" or "doing") on a given population (for example, assigning red wine drinking), rather than observing ("seeing") the values of given variables. Manipulated probabilities are the probabilities that are implicitly used in decision theory, where the different actions under consideration are manipulations.[2]

A simple form of manipulation specifies what new density P' is assigned to some variable in a population at a given time. For example, forcing everyone in an (adult) population to drink an average of 1 glass of red wine daily from t–10 to t–5, assigns $P'(rw_{t-5} = 1) = 1$. (Since rw_{t-5} measures red wine drinking for the past 5 years, an intervention on rw_{t-5} begins at t–10.) After this density has been assigned, there is a resulting joint density for the random variables at time t, denoted by $P(sex, bmi_{t\text{-}5}, ha_{t\text{-}5}, rw_{t\text{-}5}, bmi_t, ha_t, rw_t \mid\mid P'(rw_{t-5} = 1) = 1)$, where the double bar indicates the density that has been assigned to rw_{t-5}, in this case that everyone has been assigned the value

2. The use of manipulated probability densities in decision theory is often not explicit. The assumption that the density of states of nature are independent of the actions taken (act-state independence) is one way to ensure that the manipulated densities that are needed are equal to observed conditional densities that can be measured.

$rw_{t-5} = 1$.[3] This is in contrast to the conditional density $P(sex, bmi_{t\text{-}5}, ha_{t\text{-}5}, rw_{t\text{-}5}, bmi_t, ha_t, rw_t \mid rw_{t-5} = 1)$, which is the density of the variables in the subpopulation where $rw_{t-5} = 1$ because people have been observed to drink that amount of red wine, as in the unmanipulated population.

$P(sex, bmi_{t\text{-}5}, ha_{t\text{-}5}, rw_{t\text{-}5}, bmi_t, ha_t, rw_t \mid\mid P'(rw_{t-5} = 1) = 1)$ is a density, so it is possible to form marginal and conditional probability densities from it. For example, $P(ha_t \mid bmi_{t-5} = 25 \mid\mid P'(rw_{t-5} = 1) = 1)$ is the probability of having had a heart attack between t–5 and t among people who have a *bmi* of 25 at t–5, everyone having been assigned to drink an average of 1 glass of red wine daily between t–10 and t–5. In this paper, in order to simplify the exposition, it will be assumed that all attempted manipulations are successful; that is, if $P'(rw_{t-5} = 1) = x$ then $P(rw_{t-5} = 1 \mid\mid P'(rw_{t-5} = 1) = x) = x$ (that is, if rw_{t-5} is manipulated to have value 1 with probability x, then in the manipulated population, rw_{t-5} has value 1 with probability x.) For example, if it is assumed that $P'(rw_{t-5} = 1) = 1$ then $P(rw_{t-5} = 1 \mid\mid P'(rw_{t-5} = 1) = 1) = 1$, that is if everyone has been assigned to drink an average of 1 glass of red wine per day for 5 years (denoted $P'(rw_{t-5} = 1) = 1$), that everyone has done so.

In a randomized trial, a manipulation could set $P'(rw_{t-5} = 1) = 0.5$ and $P'(rw_{t-5} = 0) = 0.5$, in which case the resulting density is $P(sex, bmi_{t\text{-}5}, ha_{t\text{-}5}, rw_{t\text{-}5}, bmi_t, ha_t, rw_t \mid\mid \{P'(rw_{t-5} = 1) = 0.5, P'(rw_{t-5} = 0) = 0.5\})$.

In more complex manipulations, different probabilities can be assigned to different subpopulations. For example, the amount of red wine someone is assigned to drink could be based on *sex*: $P'(rw_{t-5} = 0 \mid sex = 0) = 0.25$, $P'(rw_{t-5} = 1 \mid sex = 0) = 0.75$, $P'(rw_{t-5} = 0 \mid sex = 1) = 0.\ 5$, $P'(rw_{t\text{-}5} = 2 \mid sex = 1) = 0.5$. The resulting density is $P(sex, bmi_{t\text{-}5}, ha_{t\text{-}5}, rw_{t\text{-}5}, bmi_t, ha_t, rw_t \mid\mid \{P'(rw_{t-5} = 0 \mid sex = 0) = 0.25, P'(rw_{t-5} = 1 \mid sex = 0) = 0.75, P'(rw_{t-5} = 0 \mid sex = 1) = 0.5, P'(rw_{t\text{-}5} = 2 \mid sex = 1) = 0.5\})$. In general, which manipulations are performed on which subpopulations can be a function both of the values of various random variables, and of what other past manipulations have been performed.

In many cases the values of some variables in the pre-manipulation density are stable, and the temporal indices on those variables are omitted. Similarly, if it is assumed that variables in the post-manipulation population eventually stabilize to fixed values, the time indices of those variables are omitted in the post-manipulation density, and the time-independent variables refer to the stable values. Both of these kinds of omissions of time indices are illustrated by the use of *sex* in the example.

In contrast to conditional probabilities, which can be estimated from samples from a population, typically the gold standard for estimating manipulated densities is an experiment, often a randomized trial. However, in many cases experiments are too expensive, too difficult, or not ethical to carry out. This raises the question of what can be determined about manipulated probability densities from samples from a population, possibly in combination with a limited number of randomized trials. The problem is even more difficult because the inference is made from a set of measured random variables $\mathbf{O}$ from samples that might not contain variables that are causes of multiple variables in $\mathbf{O}$.

With causal inference, as with statistical inference, it is generally the case that in order to make inference tractable both computationally and statistically, simplifying assumptions are made. One kind of simplifying assumption common to both statistical and causal inference is the assumption that the population distribution lies in some parametric family (for example, Gaussian) or that relationships between variables are

3. There is no completely standard notation for denoting a manipulated density. This notation is adapted from Lauritzen (1999).

exactly linear. An example of a simplifying assumption unique to causal inference is that multiple causal mechanisms relating variables do not exactly cancel (Section 3). So, although the goal of Problem 2 is stated as finding a consistent estimate of a manipulated density, it is more realistic to state the goal as finding a sufficiently good estimate of a manipulated density when the sample size is large enough.

Problem 2 is usually broken into two parts: finding a set of causal models from sample data, some manipulations (experiments) and background assumptions (Sections 3 and 4), and predicting the effects of a manipulation from a set of causal models (Section 3). Here, a "causal model" (Section 3) specifies for each possible manipulation that can be performed on the population (including the manipulation that does nothing to a population) a post-manipulation density over a given set of variables. In some cases, the inferred causal models may contain unmeasured variables as well as measured variables.

In analogy to the goals of statistical modeling, it would be more accurate but much more vague to state that the goal in Problem 3 is to find a useful (for example, sufficiently simple, sufficiently accurate, etc.) causal model, rather than a true causal model.

The reason that the stated goal for the output of Problem 3 is a set of causal models, is that it is generally not possible to reliably find a true causal model given the inputs. Furthermore, in contrast to predictive models, even if a true causal model can be inferred from a sample from the unmanipulated population, it generally cannot be validated on a sample from the unmanipulated population, because a causal model contains predictions about a manipulated population that might not actually exist. This has been a serious impediment to the improvement of algorithms for constructing causal models, because it makes evaluating the performance of such algorithms difficult. It is possible to evaluate causal inference algorithms on simulated data, to employ background knowledge to check the performance of algorithms, and to conduct limited (due to expense, time, and ethical constraints) experiments, but these serve as only partial checks how algorithms perform on real data in a wide variety of domains. For examples, see the Causality Challenge (http://www.causality.inf.ethz.ch/challenge.php).

In the Special Topic on Causality in this journal, Shpitser and Pearl (2008) and Zhang (2008) address Problem 4. Bromberg and Margaritis (2009), Pellet and Elisseeff (2008), He and Geng (2009), and (indirectly) Kang and Tian (2009), Aliferis et al. (2010a), and Aliferis et al. (2010b) address Problem 3. Both the problems and the papers will be described in more detail in subsequent sections.

2.3. Effects of Counterfactual Manipulations

There are cases in ethics, the law, and epidemiology in which there are questions about applying a manipulation to a subpopulation whose membership cannot be measured at the time that the manipulation is applied. For example, epidemiologists sometimes want to know what would the effect on heart attacks have been, if a manipulation such as assigning moderate drinking of red wine from t–10 to t–5, had been applied to the subpopulation which has *not* moderately drunk red wine from t–10 to t–5. When the manipulation under consideration assigns a value to a random variable to a subpopulation with a different actual value of the random variable, the probability in question is a *counterfactual* probability. If the subpopulation that did not moderately drink red wine between t–10 and t–5 differs systematically from the rest of the population with respect

to causes of heart attacks, the subpopulations' response to being assigned to drink red wine would be different than the rest of the population.

Questions about counterfactual probabilities arise naturally in assigning blame in ethics or in the law. For example, the question of whether tobacco companies were negligent in the case of someone who smoked and developed lung cancer depends upon the probability that person would not have gotten lung cancer if they had not smoked.

A counterfactual probability cannot be estimated directly from a randomized experiment, because it is impossible to perform a randomized experiment that assigns moderate red wine drinking between t–10 to t–5 to a group of people who already have not been moderate wine drinkers between t–10 and t–5. This raises the question of how counterfactual probabilities can be estimated. One general approach is to assume that the value of red wine drinking between t–10 and t–5 contains information about hidden causes of red wine drinking that are also causes of heart attacks.

In the Special Topic on Causality in this journal, Shpitser and Pearl (2008) describes a solution to Problem 5 in the case where the causal graph is known, but may contain unmeasured common causes.

3. Causal Models

This section describes several different kinds of commonly used causal models, and how to use them to calculate the effects of manipulations. The next section describes search algorithms for discovering causal models.

A (parametric) *statistical model* (with free parameters) is a set of probability densities that can be mapped into a single density by specifying the values of the free parameters (for example, a family of multivariate Gaussian densities).[4] For example, a Hidden Markov Model with a fixed structure but free parameters is a statistical model that represents a certain set of probability densities. A *causal model with free parameters* also specifies a set of probability densities over a given set of variables; however, in addition, for each manipulation that can be performed on the population it also specifies a set of post-manipulation probability densities over a given set of variables. A causal model with free parameters together with the values of the free parameters is a *causal model with fixed parameters*; a causal model with fixed parameters is mapped to a single density given a specification of a manipulation.

Often, a causal model is specified in two parts: a statistical model, and a causal graph that describes the causal relations between variables. The most frequently used causal models belong to two broad families: (1) causal Bayesian networks, (2) structural equation models. Causal Bayesian networks (and related models), specify a density for a variable as a function of the values of its causes. Structural equation models (SEMs) specify the value of a variable as a function of the values of its causes (typically including some unmeasured noise terms.) However, not surprisingly, the two kinds of models are closely linked, as explained in Section 3.2.

The statistical setup for both causal Bayesian networks and structural equation models is a standard one. There is a population of units, where depending upon the problem, the units could be people, cities, cells, genes, etc. It is assumed that there is a density over the population, which assigns probabilities to each measurable subset

4. In the nomenclature of machine learning, what this article calls a "model (with free parameters)" is often called a "model family" or "learning machine" and a "model (with fixed parameter values)" is often called a "model instance" or "model".

Problem 1: Predictive Modeling

Input: Samples from a density $P(\mathbf{O})$ (where $\mathbf{O}$ is a set of observed random variables), and two sets of variables $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{O}$.
Output: A consistent, efficient estimate of $P(\mathbf{Y} \mid \mathbf{X})$.

Problem 2: Causal Predictive Modeling

Input: Samples from a population with density $P(\mathbf{O})$, and a (possibly empty) set of manipulated densities $P(\mathbf{O} \mid\mid M_1), \ldots\ P(\mathbf{O} \mid\mid M_n)$, a manipulation M, and sets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{O}$.
Output: A consistent, efficient estimate of $P(\mathbf{Y} \mid \mathbf{X} \mid\mid M)$ if possible, and an output of "not possible" otherwise.

Problem 3: Constructing Causal Models from Sample Data

Input: Samples from a population with density $P(\mathbf{O})$, a (possibly empty) set of manipulated densities $P(\mathbf{O} \mid\mid M_1), \ldots\ P(\mathbf{O} \mid\mid M_n)$, and background assumptions.
Output: A set of causal models that is as small as possible, and contains a true causal model that contains at least the variables in $\mathbf{O}$.

Problem 4: Predicting the Effects of Manipulations from Causal Models

Input: An unmanipulated density $P(\mathbf{O})$, a set $\mathbf{C}$ of causal models that contain at least the variables in $\mathbf{O}$, a manipulation M, and sets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{O}$.
Output: A function g such that $P(\mathbf{Y} \mid \mathbf{X} \mid\mid M) = g(P(\mathbf{O}), \mathbf{C}, M, \mathbf{X}, \mathbf{Y})$ if one exists, and an output of "no function" otherwise.

Problem 5: Counterfactual predictive modeling

Input: An unmanipulated density $P(\mathbf{O})$, a set $\mathbf{C}$ of causal models that contain at least the variables in $\mathbf{O}$, a counterfactual manipulation M, and sets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{O}$.
Output: A function g such that $P(\mathbf{Y} \mid \mathbf{X} \mid\mid M) = g(P(\mathbf{O}), \mathbf{C}, M, \mathbf{X}, \mathbf{Y})$ if one exists, and an output of "no function" otherwise.

(event) of the population. Each unit also has a set of properties at a time, where the properties are represented by random variables, which are functions from the units to real numbers. The following sections describe the causal part of the model.

3.1. Causal Bayesian Networks

A *Bayesian network* is a pair $\langle G, P \rangle$, where G is a directed acyclic graph (DAG) whose vertices are random variables, and P is a density such that each variable V in G is independent of variables that are neither descendants nor parents of V in G,[5] conditional on the parents of V in G. In this case P is said to satisfy the *local directed Markov condition* for G.

There are two conditions that are equivalent to the local directed Markov condition described below that are useful in causal inference: the global directed Markov condition, and factorization according to G, both of which are described next.

The conditional independence relations specified by satisfying the local directed Markov condition for DAG G might also entail other conditional independence relations. There is a fast algorithm for determining from G whether a given conditional independence relation is entailed by satisfying the local directed Markov condition for G, that uses the d-separation relation, a relation among the vertices of G. A variable B is a *collider* (*v-structure*) *on a path* U if and only if U contains a subpath $A \rightarrow B \leftarrow C$. For disjoint sets of vertices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ in a DAG G, $\mathbf{X}$ is *d-connected* to $\mathbf{Y}$ given $\mathbf{Z}$ if and only if there is an acyclic path U between some member X of $\mathbf{X}$, and some member Y of $\mathbf{Y}$, such that every collider on U is either a member of $\mathbf{Z}$ or an ancestor of a member of $\mathbf{Z}$, and every non-collider on U is not in $\mathbf{Z}$.[6] For disjoint sets of vertices, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, $\mathbf{X}$ is *d-separated* from $\mathbf{Y}$ given $\mathbf{Z}$ if and only if $\mathbf{X}$ is not d-connected to $\mathbf{Y}$ given $\mathbf{Z}$. $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in DAG G if and only if $\mathbf{X}$ is independent of $\mathbf{Y}$ conditional on $\mathbf{Z}$ in every density that satisfies the local directed Markov condition for G (Pearl, 1988). If $\mathbf{X}$ is independent of $\mathbf{Y}$ conditional on $\mathbf{Z}$ in P whenever $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{Z}$ in G, then P satisfies the *global directed Markov condition* for G.

For the set of random variables $\mathbf{V}$ in G, a density $P(\mathbf{V})$ *factors according to* DAG G iff

$$P(\mathbf{V}) = \prod_{V \in \mathbf{V}} P(V \mid \mathbf{Parents}(V, G))$$

where **Parents**(V,G) is the set of parents of V in G.

The local directed Markov condition, the global directed Markov condition, and factorization according to a DAG G are all equivalent under mild regularity assumptions (Lauritzen et al., 1990).

A DAG can also be used to represent causal relations between variables. A is a *direct cause* of B relative to a set of variables $\mathbf{V}$ in a population when there exist two manipulations of $\mathbf{V} \backslash \{B\}$ (that is, all the variables in $\mathbf{V}$, except B, are manipulated to specific values) that differ only in the values assigned to A and that produce different probability densities of B. A *causal DAG* G for a population contains an edge $A \rightarrow B$ iff A is a direct cause of B in the specified population.

In order to use samples from probability densities to make causal inferences, some assumptions relating causal relations to probability densities need to be made. The

5. X is a *parent* of Y if the graph contains the edge $X \rightarrow Y$. Y is a *descendant* of X if there is a directed path from X to Y.

6. For both the d-separation relation and the independence relation, if $\mathbf{X}$ contains a single vertex X, then X will be written instead of $\{X\}$, and similarly for $\mathbf{Y}$ and $\mathbf{Z}$. D-connection can also be defined for cyclic graphs and graphs with double-headed arrows (Spirtes, 1995; Koster, 1999; Cox and Wermuth, 1996).

following Causal Markov Assumption is commonly made, if only implicitly. A set of variables **V** is *causally sufficient* iff there is no variable C not in **V** that is a direct cause of more than one variable in **V** (relative to $\mathbf{V} \cup \{C\}$).

Causal Markov Assumption: For a causally sufficient set of variables **V** in a population N with density $P(\mathbf{V})$, $P(\mathbf{V})$ satisfies the local directed Markov condition for the causal DAG of N.

Under the Causal Markov Assumption, in a causal Bayesian network a manipulation of X to $P'(X \mid \mathbf{Y})$ (where **Y** is assumed to contain only non-descendants of X in a causal DAG G) simply replaces the term $P(X \mid \mathbf{Parents}(X,G))$ in the factorization of the joint density by the manipulated density $P'(X \mid \mathbf{Y})$:

$$P(\mathbf{V}||P'(X \mid \mathbf{Y})) = P'(X \mid \mathbf{Y}) \prod_{V \in \mathbf{V} \setminus \{X\}} P(V \mid \mathbf{Parents}(V, G)).$$

This is called the *manipulation rule*. The importance of the manipulation rule is that if the causal DAG is known, and the unmanipulated density can be estimated from a sample, it allows the prediction of the effect of an unobserved manipulation. Hence the manipulation rule is the solution to Problem 4, in the special case where the observed variables are causally sufficient, and the unique correct causal DAG is known.

The solution to Problem 4 is more difficult when the set of observed variables is not causally sufficient. There are sufficient and (almost) necessary rules for determining which manipulated conditional probability densities are invariant under a given manipulation (that is, which densities are the same in the unmanipulated population and the manipulated population) and rules for how to express some non-invariant conditional densities as functions of observed densities (Spirtes et al., 1993). Pearl's do-calculus extended the sufficient and (almost) necessary conditions for determining which conditional densities were invariant from single manipulations to sequences of manipulations, and showed how a broader range of non-invariant manipulated densities could be expressed in terms of observed densities (Pearl, 1995). In the Special Topic on Causality of this journal, Shpitser and Pearl (2008) describe an algorithm that has recently been developed and show that it is a complete solution to Problem 4 in the special case where a unique causal DAG is known (Shpitser and Pearl, 2006a,b; Huang and Valtorta, 2006).

Calculation of the effect of a counterfactual manipulation when causal sufficiency does not hold among the observed variables is a complex operation that requires several copies of the causal graph in order to keep track both of the actual value of the variable being manipulated, and the counterfactual value of the variable being manipulated. In the Special Topic on Causality, Shpitser and Pearl (2008) describe for the first time an algorithm that is a complete solution to Problem 5 in the special case where a unique causal DAG is known, even if the set of observed variables is not causally sufficient.

3.2. Structural Equation Models (SEMs)

Structural equation models are widely used in the social sciences (Bollen, 1989) and in some natural sciences. The set of random variables in a structural equation model (SEM) can be divided into two subsets, the "error variables" or "error terms," and the substantive variables (for which there is no standard terminology in the literature). The substantive variables are the variables of interest, but they are not necessarily all observed. Which variables are substantive, and which variables are error terms can vary with the analysis of the problem. Each substantive variable is a function of other

substantive variables and a unique error term. The joint density over the substantive variables is a function of the density over the error terms and of the functions relating each variable to its causes. There is an edge $A \rightarrow B$ in the graph ("path diagram") of a SEM when A is a non-trivial argument in the function for B. A manipulation of a variable B to a constant c is represented in a SEM by replacing the equation for B with $B = c$.

In general, the graph of a SEM may have cycles (that is, directed paths from a variable to itself), and may explicitly include error terms with double-headed arrows between them to represent that the error terms are dependent (for example, $\varepsilon_A \leftrightarrow \varepsilon_B$); if no such double-headed edge exists in the graph, the error terms are assumed to be independent of each other. An error term is not explicitly included in the graph unless it is the endpoint of a double-headed arrow; otherwise, an error term occurs in the SEM model, but is not shown in the graph. If the graph has no directed cycles and no double-headed arrows, then the graph is a DAG and the SEM is said to be *recursive*; otherwise it is said to be *non-recursive*.

In a recursive SEM, if the marginal density over the substantive variables is $P(\mathbf{V})$, then $\langle G, P(\mathbf{V})\rangle$ is a Bayesian network (Spirtes et al., 2001; Pearl, 2000); for short, say that a SEM with an associated graph that is a DAG is a Bayesian network (although the SEM contains some extra structure in that it entails that any non-determinism among the substantive variables is only due to the marginalization of the error terms.)

Non-recursive SEMs are of interest because they allow for the representation of feedback (with cycles) or unmeasured common causes (represented by double-headed arrows.) In the case of linear non-recursive SEMs, it is still possible to deduce the conditional independencies (or more generally the zero partial correlations) entailed for all Gaussian SEMs (or more generally linear SEMs) from the graph G of a non-recursive linear SEM using a minor modification of the d-separation relation (Koster, 1999; Spirtes, 1995).

For both theoretical interest and for the purposes of efficient (constraint-based) search of the space of linear non-recursive SEMs without cycles (Section 4.2), it is of interest to find some proper subset of the set of all conditional independence relations entailed by the (modified) d-separation which entail all the rest, that is, a modified form of the local directed Markov condition. (In contrast to the recursive case, where such a subset is given by the independencies entailed by the local directed Markov condition, in the non-recursive case SEMs do not generally satisfy the local directed Markov condition). One such subset of conditional independencies was described by Richardson (2003). In this special issue, the paper by Kang and Tian (2009) describes another such subset, which is often smaller than the one described by Richardson, and hence might be more useful for the purposes of search.

4. Model Search

Traditionally, there have been a number of different approaches to causal discovery. The gold standard of causal discovery has typically been to perform planned or randomized experiments (Fisher, 1971). There are obvious practical and ethical considerations that limit the application of experiments in many instances, particularly on human beings. Moreover, recent data collection techniques and causal inference problems raise several practical difficulties regarding the number of experiments that need to be performed in order to answer all of the outstanding questions.

In the absence of experiments, in practice (particularly in the social sciences) search for causal models is often informal, and based on a combination of background assumptions about causal relations together with statistical tests of the causal models. If a model is rejected by a statistical test, the researcher looks for a modification of the original hypothesized model that will pass a statistical test. The search typically halts when a model that is compatible with background knowledge does not fail a statistical test (Rodgers and Maranto, 1989). Often, the final model is presented, and the search itself is not described. Informal searches of this kind fail to account for multiple testing problems, and can potentially lead to severe overfitting problems. The reliability of such a search depends upon the correctness of the background assumptions, and the extent to which the space of alternatives compatible with the background assumptions was searched. Furthermore, unless the background assumptions are very extensive, or the number of variables is tiny, it is not feasible to estimate and test all of the models compatible with background assumptions. This is further complicated by the fact that, as explained below, for reliable causal inference it is not sufficient to find one model that passes a statistical test; instead it is necessary to find all such models. Recent developments in automated model search have attempted to address these problems with traditional methods of search.

There are several major differences between model search in the case of predicting the unmanipulated value of Y, and model search in the case of predicting the post-manipulation value of Y, based on the different uses of statistical models and causal models described in the following section.

4.1. Underdetermination of Causal Models by Data

Causal model (with fixed parameter) search is often broken into two parts: search for a causal graph, and estimation of the free parameters from sample data and the causal graph. (In some cases, the prediction of the effects of manipulations does not require estimating all of the free parameters, but does require estimating functionals of the free parameters.) Generally, the estimation of the free parameters employs standard statistical methods. For example, in a linear SEM with a recursive DAG, no unmeasured variables, and Gaussian errors, the maximum likelihood estimate of the edge coefficients is given by regressing each variable on its parents in the DAG. This section concentrates on the search for causal graphs, because the search for causal graphs is significantly different than the search for graphs that are to be used only as statistical models.

In causal model search based on unmanipulated data, if no preference for simpler models over more complex models is made, then the causal models are underdetermined to such an extent that useful causal inference is impossible for many important parametric families (for example, Gaussian or multinomial) or unrestricted probability densities. There are a variety of simplicity assumptions that select simpler models over more complex models that can be made. In the case of search based upon maximizing some model score given sample data (such as the Bayesian Information Criterion), the simplicity assumption is a penalty for complexity built into the score (Chickering, 2002). For search that is not based upon model scores, the following simplicity assumption is often, if implicitly made:

Causal Faithfulness Assumption: For a causally sufficient set of variables $\mathbf{V}$ in a population N, every conditional independence relation true in the density over $\mathbf{V}$ is entailed by the local directed Markov condition for the causal DAG of N.

There are several other versions of the assumption that are considerably weaker than the one stated here (and more intuitively justifiable) but still permit reliable causal inference, at the cost of requiring more complicated algorithms with more complex and somewhat less informative output (Ramsey et al., 2006).

However, even given the Causal Markov and Faithfulness Assumptions and the assumption that the observed variables are causally sufficient, the true causal model is underdetermined by the available evidence and background assumptions, because of the hierarchy of equivalence relations described below.

Two different DAGs G and G' that have the same set of d-separation relations are said to be *Markov* (*conditional independence*, *d-separation*) *equivalent*.

For each DAG G, there is a set $\mathbf{P}$ of probability densities that satisfy the local directed Markov condition for G, denoted $\mathbf{P}(G)$ that are said to be *represented* by G. In many cases, some subset of $\mathbf{P}$ that belongs to a parametric or semi-parametric family $\mathbf{F}$ is of interest; for example, the Gaussian subset of $\mathbf{P}$. Two DAGs G and G' are *statistically equivalent with respect to* $\mathbf{F}$ iff $\mathbf{P}(G) \cap \mathbf{F} = \mathbf{P}(G') \cap \mathbf{F}$. Two DAGs that are statistically equivalent with respect to $\mathbf{F}$ are the same statistical model with respect to $\mathbf{F}$.

Two DAGs are *causally equivalent* (with respect to a family of densities $\mathbf{F}$) iff they represent the same set of probability densities (in family $\mathbf{F}$) for every manipulation (including the null manipulation.) It is easy to see that no pair of DAGs that differ in their structure can be causally equivalent.

As an example, $A \rightarrow B \leftarrow C \leftarrow D$ and $A \rightarrow B \leftarrow C \rightarrow D$ are Markov equivalent, but not causally equivalent. They are statistically equivalent with respect to Gaussian SEMs, but they are not statistically equivalent with respect to linear SEMs with at most one Gaussian error term, and no determinism among the substantive variables (Shimizu et al., 2006).[7]

In the absence of further information (for example, samples from manipulated densities or background domain knowledge) all of the DAGs in a statistical equivalence class fit the data and the background assumptions equally well, and are equally simple. Hence standard scores such as Bayesian Information Criterion, Minimum Description Length, chi-squared statistics, etc. all produce equal scores for the alternative DAGs in a statistical equivalence class for all data sets -- in general, there is no one DAG with the highest score, but rather, there is a set of DAGs with equally high scores. Furthermore, for computational and statistical reasons, it is sometimes easier to search for the Markov equivalence class of DAGs, even if it is known that the statistical equivalence class is a proper subset of the Markov equivalence class.

If the DAG is to be used to estimate observational (not manipulated) conditional densities, this is not a problem, because all of the statistically equivalent models will produce the same estimate. However, if the DAG is to be used to predict the effects of manipulations, then the different models will make different predictions about at least some manipulations. So in the case of causal modeling, unlike observational statistical modeling, it is not enough to simply output one arbitrarily selected DAG from a set of highest scoring DAGs -- it is important to output the entire set, so that all of the different answers given by the different models can be taken into account. Once the set of highest scoring DAGs is found, the problem of dealing with the underdetermination of the effects of manipulations must also be dealt with. These problems are described in more detail in the next two subsections.

7. In a linear SEM it is assumed that each variable is a linear function of its causal parents and a unique error term; in a Gaussian SEM it is assumed in addition that the errors term are Gaussian.

If the assumption of causal sufficiency of the observed variables is not made, all three kinds of equivalence classes have corresponding equivalence classes over the set of observed variables, and the problem of causal underdetermination becomes much more severe. For example, for a given set of observed variables **O**, the Markov equivalence class relative to **O** consists of the set of all DAGs (possibly containing variables not in **O**) that have the same set of d-separation relations among the variables in **O**; this might be much larger than the Markov equivalence class that consists of the set of DAGs (containing only variables in **O**) that have the same set of d-separation relations among the variables in **O**.

4.2. Constraint-based Search

First, the problem where only sample data from the unmanipulated population density is available will be considered. The number of DAGs grows super-exponentially with the number of vertices, so even for modest numbers of variables it is not possible to examine each DAG to determine whether it is compatible with the population density given the Causal Markov and Faithfulness Assumptions. Constraint based search algorithms, given as input an oracle that returns answers about conditional independence in the population and optional background knowledge about orientations of edges, return a representation of a Markov equivalence class (or if there is background knowledge, a subset of a Markov equivalence class) on the basis of oracle queries. One example of a constraint-based algorithm is the PC algorithm (Spirtes and Glymour, 1991). If the oracle always gives correct answers, and the Causal Markov and Causal Faithfulness Assumptions hold, then the PC algorithm always outputs a Markov equivalence class that contains the true causal model, even though the algorithm does not check each directed acyclic graph. In the worse case, it is exponential in the number of variables, but for sparse graphs it can run on hundreds of variables in an acceptable amount of time (Spirtes and Glymour, 1991; Spirtes et al., 1993; Meek, 1995). Kalisch and Buhlmann (2007) showed that under a strengthened version of the Causal Faithfulness Assumption, the PC algorithm is uniformly consistent for very high-dimensional, sparse DAGs where the number of nodes is allowed to quickly grow with sample size n, as fast as $O(n^a)$ for any $0 < a < \infty$. In practice, the judgments about conditional independence are made by performing (fallible) statistical tests. A number of other variants of constraint-based algorithms have been proposed that improve on either the accuracy or speed of the PC algorithm, or to weaken the assumptions under which it is guaranteed to be correct.

There are both advantages and disadvantages of constraint based searches as compared to either a Bayesian approach to the problem of causal discovery (Heckerman and Geiger, 1995), or an approach based upon assigning a score to each causal model for a given data set (for example, Bayesian information criterion) and searching for the set of causal models that maximize the score (Chickering, 2002).

The disadvantages of constraint-based search include that the output of constraint-based searches give no indication of how much better the best set of output models is compared to the next best set of models; at small sample sizes tests of conditional independence have low power, particularly when many variables are conditioned on; mistakes made early in a constraint based searches can lead to later mistakes; and if the only constraints used are conditional independence constraints, as is often but not always the case, then at best the search outputs a Markov equivalence class, rather

than a statistical equivalence class.[8] In addition, constraint-based methods have the problem of multiple testing. If no control is made for multiple testing, the models may overfit the data. However, adjustments to control for overfitting, such as the Bonferroni correction, are often too conservative and as a result the corrected statistical tests are not very powerful. The issue of multiple testing appears in Bayesian approaches to causal discovery as multiple causal model scoring. The issue is handled automatically by Bayesian methods by their use of prior probabilities (Heckerman et al., 1999).

The advantages of constraint-based algorithms are that they are easier to generalize to the case where the observed variables are not causally sufficient, they are generally fast, and given recent developments of non-parametric conditional independence tests, they are applicable without parametric assumptions (Tillman et al., 2009).

In the Special Topic on Causation, Bromberg and Margaritis (2009) models the problem of low power of statistical tests as a knowledge base containing a set of independence facts related through conditional independence axioms that may contain errors due to errors in the tests of conditional independence. The inconsistencies are resolved through the use of a defeasible logic called argumentation that is augmented with a preference function. The logic is used to reason about and possibly correct errors in these tests. Experimental evaluation shows significant improvements in the accuracy of argumentative over purely statistical tests, and improvements on the accuracy of causal structure discovery from sampled data from randomly generated causal models and on real-world data sets.

The contributions to the Special Topic on Causality by Aliferis et al. (2010a) and Aliferis et al. (2010b) show that a general framework for localized causal membership structure learning is very accurate even in small sample situations and can thus be used as a first step for efficient global structure learning, as well as accurate prediction and feature selection. It also provides extensive empirical comparisons of state of the art causal learning methods with non-causal methods for the above tasks. In addition, they show that unexpectedly some constraint-based methods are self-correcting with respect to multiple testing, and this may constitute a new methodology for control of multiple statistical testing.

Another problem with constraint-based algorithms is to make them feasible for even higher dimensional data sets. In the Special Topic on Causality, Pellet and Elisseeff (2008) link the causal model search problem to a classic machine learning prediction problem. They show how a generic feature-selection algorithm returning strongly relevant variables can be turned into a causal model search algorithm. Under the Causal Markov and Causal Faithfulness Assumptions, the smallest set of features relevant to predicting a vertex V is the set of parents, children, and parents of children of V. Ideally, the variables returned by a feature-selection algorithm identify those features of the causal graph. Then further processing removes the extra edges (between V and those variables that are parents of children of V but that are neither parents nor children of V) and provides as many orientations as possible. This algorithm is more accurate than PC and other constraint-based algorithms, and has the advantage that it can use arbitrary feature-selection techniques developed for high-dimensional data sets under different assumptions to provide causal model learning algorithms for high-dimensional data under those assumptions.

8. For searches that use non-conditional independence constraints see Silva et al. (2006) and Shpitser et al. (2009).

4.3. Dealing with Underdetermination

One possibility for dealing with the underdetermination of causal models by observational data is to strengthen the available information by sampling from manipulated densities, or in other words, performing experiments.

In the Special Topic on Causality, He and Geng (2009) propose an algorithm for distinguishing between members of a Markov equivalence class by a set of optimally designed experiments. They consider several kinds of experiments, and both a batch-design and a sequential design to minimize the required number of manipulations using both minimax and maximum entropy criteria.

If some members of the Markov equivalence class cannot be eliminated through experimentation, there are several different approaches to using the entire Markov equivalence class to predict the effects of manipulations. (This is Problem 4 in the case where the predictions are made from a set of causal models **C** rather than a single causal model, and the set of observed variables may not be causally sufficient.) One possibility is to predict an interval for the potential effects of the manipulated quantity, instead of a point value. Theoretically, an interval could be obtained by calculating the manipulated quantity for each DAG G in the Markov equivalence class, and taking the lower and upper limits. Depending upon how many different SEMs there are in the output, this is sometimes computationally feasible (Maathuis et al., 2009).

A second possibility is to use a Bayesian approach, and perform model averaging. That is, a prior probability is placed over each causal DAG G, and a posterior probability for each G is calculated. Then the manipulated quantity is calculated for each G in the output of the search, and the results are averaged together. This requires putting a prior probability over each graph; in addition, if there are many graphs in the output, then this may not be computationally feasible (Hoeting et al., 1999).

A third alternative is to have an algorithm that determines whether each DAG in the Markov equivalence class predicts the same effect of a given manipulation. For example, if the Markov equivalence class contains $A \rightarrow B \leftarrow C \rightarrow D$ and $A \rightarrow B \leftarrow C \leftarrow D$, then the two causal DAGs disagree about the effect of manipulating D on C, but agree about the effect of manipulating A on B. Even when the observed variables are not causally sufficient there is an algorithm (the Prediction Algorithm) for determining when all of the DAGs in a Markov equivalence class relative to the observed variables agree about the effect of a particular manipulation, and returns the common value of the predicted manipulation when they do all agree (Spirtes et al., 1993). However, this algorithm is known to be correct but incomplete (that is, it sometimes returns "don't know" even when all models in the equivalence class agree on the effect of a particular manipulation). In this special issue, Zhang (2008) provides a modified version of Pearl's do-calculus that is more complete than the Prediction algorithm.

5. Open Questions

The following is an overview of important problems that remain in the domain of causal modeling.

1. Matching causal models and search algorithms to causal problems. There are a wide variety of causal models that have been employed in different disciplines. What new models and search algorithms are appropriate for different domains such as feedback or reversible systems (Richardson, 1996)? What search algorithms are appropriate for different combinations of kinds of data, such as experimental

and observational data (Eberhardt et al., 2005; Cooper and Yoo, 1999; Yoo and Cooper, 2004; He and Geng, 2009)? What search algorithms are appropriate for different kinds of background knowledge, and different families of probability densities?

2. Model selection, and prior knowledge. What kind of scores can be used to best evaluate causal models from various kinds of data? In a related vein, what are good families of prior densities that capture various kinds of background knowledge?

3. Improve efficiency and efficacy of search algorithms. How can search algorithms be improved to incorporate different kinds of background knowledge, search over different classes of causal models, run faster, handle more variables and larger sample sizes, be more reliable at small sample sizes, and produce output that is as informative as possible?

4. Characterization of search algorithms. For causal search algorithms, what are their semantic and syntactic properties (for example, soundness, consistency, maximum informativeness)? What are their statistical properties (pointwise consistency, uniform consistency, sample efficiency)?[9] What are their computational properties (computational complexity)?

5. Adding or relaxing simplifying assumptions. What plausible alternatives are there to the Causal Markov and Faithfulness Assumptions? Are there other assumptions that might be weaker and hold in more domains and applications without much loss about what can be reliably inferred? Are there stronger assumptions that are plausible for some domains that might allow for stronger causal inferences? How often are these assumptions violated, and how much do violations of these assumptions lead to incorrect inferences? Can various statistical assumptions be relaxed? For example, what if the sample selection process is not i.i.d., but may be causally affected by variables of interest?

6. Derivation of consequences from causal graph and unmanipulated densities. Shpitser and Pearl have given complete algorithms for deriving the consequences of various causal models with hidden common causes in terms of the unmanipulated density and the given manipulation (Shpitser and Pearl, 2008). Partial extensions of these results to deriving consequences from sets of causal models have been given (Zhang, 2008); are there further extensions to derivations from sets of causal models?

7. New constraints for structure learning. The Causal Markov and Causal Faithfulness Assumptions, in addition to entailing conditional independence constraints on densities, also entail other constraints on densities. For example, in a linear SEM, if an unobserved variable T causes observed variables X_1, X_2, X_3, X_4, and there are no other causal relations among these variables, then there are no entailed conditional independence relations among just the observed variables X_1, X_2, X_3,

9. Intuitively, an estimator is pointwise consistent when as the sample size increases without limit, regardless of the true value, with probability 1 the absolute value of the difference between the estimator and the true value approaches zero. An estimator is uniformly consistent if for any given ϵ and δ, there is a single sample size such that in the worst case, the probability is less than ϵ that the absolute value of the difference between the estimator and the true value is greater than δ. For precise definitions in the causal context, see Robins et al. (2003).

X_4. However, the SEM entails $\text{cov}(X_1,X_2)\ \text{cov}(X_3,X_4) = \text{cov}(X_1,X_3)\ \text{cov}(X_2,X_4) = \text{cov}(X_1,X_4)\ \text{cov}(X_2,X_3)$ regardless of the values of the free parameters. This information is useful in finding causal structure with unmeasured variables. In addition, there are sometimes constraints that are not conditional independence constraints on the density of the observed variables that do not depend upon any parametric assumptions (Shpitser et al., 2009). How can these non-parametric constraints be incorporated into search algorithms?

8. Find variable definitions. In many domains, such as fMRI research, there are thousands of variables, but the measured variables do not correspond to functional units of the brain. How is it possible to define new variables that are functions of the measured variables, but more useful for causal inference and more meaningful?

9. Find new applications of causal inference. Applications of causal inference algorithms in many domains (Cooper and Glymour, 1999) help test and improve causal inference algorithms, suggest new problems, and produce domain knowledge.

10. Creating good benchmarks. What are the most appropriate performance measures for causal inference algorithms? What benchmarks can be established? What is the best research design for testing causal inference algorithms?

11. Formal connections between different causal modeling approaches. Many different fields have studied causal discovery including Artificial Intelligence, Econometrics, Operations Research, Control Theory, and Statistics. What are the formal connections between the different models, assumptions, and algorithms used in each of these fields? What can each of these domains learn from the others?

Acknowledgments

I would like to thank Isabelle Guyon, Constantin Aliferis, Greg Cooper, and Rob Tillman for many helpful comments.

References

Constantin Aliferis, Alexander Statnikov, Ionnis Tsamardinos, Subramani Mani, and Xenophon Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification, Part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010a.

Constantin Aliferis, Alexander Statnikov, Ionnis Tsamardinos, Subramani Mani, and Xenophon Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification, Part II: Analysis and extensions. *Journal of Machine Learning Research*, 11:235–284, 2010b.

Kenneth A. Bollen. *Structural Equations with Latent Variables*. Wiley-Interscience, 1989.

Facundo Bromberg and Dimitris Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research*, 10: 301–340, 2009.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

Greg Cooper and Clark Glymour. *Computation, Causation, and Discovery*. AAAI Press, 1999.

Gregory Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In Kathryn Laskey and Henri Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 116–125, San Francisco, CA, 1999. Morgan Kauffman.

David Cox and Nanny Wermuth. *Multivariate Dependencies: Models, Analysis and Interpretation (Monographs on Statistics and Applied Probability)*. Chapman and Hall, 1996.

Frederick Eberhardt, Richard Scheines, and Clark Glymour. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In Fahiem Bacchus and Tommi Jaakkola, editors, *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 178–184, Arlington, VA, 2005. AUAI Press.

Ronald Fisher. *The Design of Experiments*. Macmillan Pub Co, 1971.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 10:2523–2547, 2009.

David Heckerman and Dan Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 274–282. Morgan Kaufman, 1995.

David Heckerman, Chris Meek, and Gregory Cooper. A Bayesian approach to causal discovery. In Greg Cooper and Clark Glymour, editors, *Computation, Causation, and Discovery*, pages 141–165. MIT Press, Cambridge, MA, 1999.

Jennifer Hoeting, David Madigan, Adrian Raftery, and Chris Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–401, 1999.

Yimin Huang and Marco Valtorta. Identifiability in causal Bayesian networks: A sound and complete algorithm. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 1149–1154, Edinboro, Scotland, 2006. AAAI Press.

Markus Kalisch and Peter Buhlmann. Estimating high dimensional directed acyclic graphs with the PC algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

Changsung Kang and Jin Tian. Markov properties for linear causal models with correlated errors. *Journal of Machine Learning Research*, 10:41–70, 2009.

Jan Koster. On the validity of the Markov interpretation of path diagrams of Gaussian structural equation models with correlated errors. *Scandinavian Journal of Statistics*, pages 413–431, 1999.

Steffen Lauritzen. Causal inference from graphical models. In D. Barnsdorf-Nielsen and C. Kluppenberg, editors, *Complex Stochastic Systems*, pages 141–165. Chapman and Hall, Baton Rouge, LA, 1999.

Steffen Lauritzen, Phil Dawid, B. Larsen, and H. Leimer. Independence properties of directed Markov fields. *Networks*, 20:491–505, 1990.

Marloes Maathuis, Markus Kalisch, and Peter Buhlmann. Estimating high-dimensional intervention effects from observational data. *Annals of Statistics*, 37(6A):3133–3164, 2009.

Chris Meek. Strong completeness and faithfulness in Bayesian networks. In Phillipe Besnard and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 411–419, Montreal, Quebec, 1995. Morgan Kaufman.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Jean-Philippe Pellet and Andre Elisseeff. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342, 2008.

Joseph Ramsey, Peter Spirtes, and Jiji Zhang. Adjacency-faithfulness and conservative causal inference. In Rina Dechter and Thomas Richardson, editors, *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 401–408, Cambridge, MA, 2006. AUAI Press.

Thomas Richardson. A discovery algorithm for directed cyclic graphs. In Eric Horvitz and Finn Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 454–462, Cambridge, MA, 1996. Morgan Kaufmann.

Thomas Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30:145–157, 2003.

James Robins, Richard Scheines, Peter Spirtes, and Larry Wasserman. Uniform consistency in causal inference. *Biometrika*, 90(3):491–515, 2003.

R. Rodgers and C. Maranto. Causal-models of publishing productivity in psychology. *J Appl Psychol*, 74(4):636–649, 1989.

Shohei Shimizu, Aapo Hyvarinen, Patrick Hoyer, and Yutaku Kano. Finding a causal ordering via independent component analysis. *Comput Stat Data An*, 50(11):3278–3293, 2006.

Ilya Shpitser and Judea Pearl. Identification of conditional intervention distributions. In Rina Dechter and Thomas Richardson, editors, *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 437–444, Cambridge, MA, 2006a. AUAI Press.

Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 1219–1226, Menlo Park, California, 2006b. AAAI Press.

Ilya Shpitser and Judea Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008.

Ilya Shpitser, Thomas Richardson, and James Robins. Testing edges by truncation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1957–1963. AAAI Press, 2009.

Ricardo Silva, Richard Scheines, Clark Glymour, and Peter Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.

Peter Spirtes. Directed cyclic graphical representations of feedback models. In Phillipe Besnard and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 491–499, Montreal, Canada, 1995. Morgan Kaufmann.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):67–72, 1991.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Spring-Verlag Lectures in Statistics, 1993.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning)*. The MIT Press, 2001.

Robert Tillman, Arthur Gretton, and Peter Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Proceedings of Advances in Neural Processing Information Systems 22*, pages 1847–1855, Vancouver, BC, 2009. Curran Associates, Inc.

Changwon Yoo and Gregory Cooper. An evaluation of a system that recommends microarray experiments to perform to discover gene-regulation pathways. *Artificial Intelligence in Medicine*, 31(2):169–182, 2004.

Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9:1437–1474, 2008.

JMLR: Workshop and Conference Proceedings 9:1295–1342, 2008 Submitted 7/07; Revised 2/08; Published 7/08

Using Markov Blankets for Causal Structure Learning

Jean-Philippe Pellet JEP@ZURICH.IBM.COM
André Elisseeff AEL@ZURICH.IBM.COM
Data Analytics Group
IBM Zurich Research Laboratory
Säumerstraße 4, CH–8803 Rüschlikon

Editor: David Maxwell Chickering

Abstract

We show how a generic feature-selection algorithm returning strongly relevant variables can be turned into a causal structure-learning algorithm. We prove this under the Faithfulness assumption for the data distribution. In a causal graph, the strongly relevant variables for a node X are its parents, children, and children's parents (or spouses), also known as the Markov blanket of X. Identifying the spouses leads to the detection of the V-structure patterns and thus to causal orientations. Repeating the task for all variables yields a valid partially oriented causal graph. We first show an efficient way to identify the spouse links. We then perform several experiments in the continuous domain using the Recursive Feature Elimination feature-selection algorithm with Support Vector Regression and empirically verify the intuition of this direct (but computationally expensive) approach. Within the same framework, we then devise a fast and consistent algorithm, Total Conditioning (TC), and a variant, TC_{bw}, with an explicit backward feature-selection heuristics, for Gaussian data. After running a series of comparative experiments on five artificial networks, we argue that Markov blanket algorithms such as TC/TC_{bw} or Grow-Shrink scale better than the reference PC algorithm and provides higher structural accuracy.

Keywords: causal structure learning, feature selection, Markov blanket, partial correlation, statistical test of conditional independence

1. Introduction

In this paper, we are interested in using concepts from the feature-selection field to help causal structure learning. Causal structure learning (Pearl, 2000; Spirtes et al., 2001) is a multivariate data-analysis approach that aims to build a directed acyclic graph (DAG) showing direct causal relations among the variables of interest of a given system. These so-called causal graphs can be used together with dedicated rules called *do*-calculus (Pearl, 1995) to predict the effect of interventions, that is, of structural changes in the data-generating process. In this sense, it differs significantly from traditional machine-learning techniques: given a set of interventions, we can predict the behavior of a set of variables whose joint probability distribution has changed since the model was trained.

Building the causal graph is a difficult task, subject to a series of assumptions, and provably correct algorithms have an exponential worst-case complexity. Identifying the exact causal graph is in general impossible. By means of non-interventional data, causal graphs can only be identified up to *observational equivalence*: only adjacencies and so-called V-structures (two independent causes leading to the same effect) can be specified exactly (Pearl, 2000, p. 19). Typical structure-learning algorithms thus return

partially directed acyclic graphs (PDAGs). These algorithms can be roughly classified into two categories: the *score-based* algorithms associate a score function with a DAG or PDAG given a training data set and perform, for instance, a greedy search in the space of DAGs or PDAGs (e.g., the GES algorithm, Chickering, 2002); the *constraint-based* algorithms look for dependencies and conditional dependencies in the data and build the causal graph accordingly. Well-known examples are the PC (Spirtes et al., 2001) or the IC (Pearl and Verma, 1991) algorithms. In an effort to get the best of both worlds, other algorithms use both conditional-independence tests and scores to build the network; MMHC (Tsamardinos et al., 2006) is such an example.

The range of data sets that the typical algorithms can deal with is restricted: not any probability distribution can be *faithfully* represented by a DAG. Faithfulness of the distribution is a well-defined condition: it guarantees the existence of a DAG, called a *perfect map*, where there is a one-to-one mapping between the graphical criterion of d-separation and conditional independence in the data.[1] Nilsson et al. (2007) discuss faithful distributions and other types of distributions with respect to properties of conditional independence. In the literature, Faithfulness is a precondition to prove correctness of the algorithms.

In practice, both existing score-based and constraint-based techniques deal primarily with discrete data sets. Score-based approaches for continuous variables are computationally expensive;[2] as for the constraint-based approaches, only the multivariate Gaussian case has been dealt with efficiently (Scheines et al., 1995). Margaritis (2005) proposed a distribution-free test of conditional independence, which is very computationally expensive and cannot be readily used with the current constraint-based algorithms for all but very small networks.

Coming from the machine-learning community, feature selection (John et al., 1994; Guyon and Elisseeff, 2003) is a common technique that aims at reducing the number of variables or features used for building more efficient or more robust models. Techniques have evolved to be able to handle nonlinear relationships between variables, redundant variables, in both discrete and continuous domains. Feature selection and causal structure learning are related by a common concept: the *Markov blanket* of a variable X is the smallest set $\mathbf{Mb}(X)$ containing all variables carrying information about X that cannot be obtained from any other variable.[3] In feature selection, this is the set of *strongly relevant* features; that is, of features which carry information about the target that cannot be obtained from any other variable (Kohavi and John, 1997). In a causal graph, this is the set of all parents, children, and spouses of X. The feature-selection task and the causal graph construction task can both be stated to some extent as Markov blanket identification tasks.

Relating feature selection and causal structure learning is not new. Several algorithms identifying the Markov blanket of a single variable with techniques inspired from causal structure learning have been proposed as the optimal solution to the feature-selection problem in the case of a faithful distribution. Tsamardinos and Aliferis (2003) show that for faithful distributions, the Markov blanket of a variable Y is exactly the set of strongly relevant features, and prove its uniqueness. They propose the Incremental Association Markov Blanket (IAMB) algorithm to determine it. With the same Faithfulness assumption, the Min-Max Markov Blanket algorithm (MMMB)

1. Conditional independence and d-separation are defined formally in Section 2.
2. Computationally tractable methods to learn Bayesian networks from continuous data exist (Fu, 2005), like Bach and Jordan (2003), but do not offer the causality-related theoretical correctness guarantees.
3. Some authors write "Markov blanket" without the notion of minimality, and use "Markov boundary" to note the smallest Markov blanket $\mathbf{Mb}(X)$. Even if defined as minimal, $\mathbf{Mb}(X)$ is generally not unique.

(Tsamardinos et al., 2003) identifies the Markov blanket of a variable Y by calling a subroutine Min-Max Parents and Children (MMPC). This subroutine finds the direct parents and children of Y with association measures and conditional-independence tests. MMPC is again called on each of these nodes to find potential spouses of Y. False positives are then discarded with conditional-independence tests. MMMB was further discussed by Peña et al. (2005), who propose AlgorithmMB, a similar approach based on scores and conditional-independence tests to retrieve $\mathbf{Mb}(Y)$. The HITON_MB algorithm (Aliferis et al., 2003) is similar in its main steps, and selects an optimal subset of the Markov blanket of a target variable given the Faithfulness assumption. Nilsson et al. (2007) also propose a theoretical algorithm for consistent identification of strongly relevant features in polynomial time for the class of strictly positive distributions. They also argue that some common backward feature-elimination algorithms like Recursive Feature Elimination (Guyon et al., 2002) are actually consistent, in the sense that they return the set of strongly relevant features in the large sample limit.[4]

These are examples of using causal structure learning or similar constraint-based techniques to help feature selection (see Guyon et al., 2007, for a review of those techniques). In this paper, we propose a framework to do the converse. We present a generic approach using the outcome of a consistent feature-selection algorithm *FS* to build an approximate structure of the true causal graph. If we assume that *FS* returns the Markov blanket of the variables, we can show how to turn this approximate result, called *moral graph* (Lauritzen and Spiegelhalter, 1988), into a provably correct PDAG depicting the causal structure. This approach is also used in the Grow-Shrink algorithm (Margaritis and Thrun, 1999), which also builds a moral graph before adjusting the local structure.

This paper contributes a generic algorithm to build a causal graph which clearly separates the Markov blanket identification and the needed local adjustments, an efficient algorithm to perform those adjustments, and two fast instances of the generic algorithm for multivariate Gaussian data sets. This is presented as follows: in Section 2, we first review the needed terms and definitions from feature selection and causality. In Section 3, we make the link from the outcome of a feature-selection algorithm to a causal graph by detailing the needed local adjustments and detail an efficient way to perform them. We directly apply it in Section 4, where we describe how we can build causal graphs using the RFE feature-selection algorithm. As this direct application is very computationally intensive, we then show our more efficient instantiations of the generic algorithm optimized for the multivariate Gaussian case, the TC and TC_{bw} algorithms. We list our experimental results in Section 5, showing through empirical evaluation that Markov blanket algorithms are more scalable and more accurate than the reference PC algorithm. We finally conclude in Section 6 and list proofs in Appendix A.

1.1. Notation

Boldface capitals designate either matrices or sets of random variables or nodes in a graph, depending on the context. $\mathbf{V}$ is the set of all variables in the analysis. Italicized capitals like X, Y, Z are random variables or nodes and elements of $\mathbf{V}$. Vectors are set in boldface lowercase, as $\mathbf{b}$ or $\mathbf{w}$; scalars in italics, as the number of samples n or the number of variables (the problem dimension) d. We indiscriminately write "variable" or "feature" to refer to any variable in the causal analysis or any node in a causal graph,

4. Actually, their definition of consistency has to do with returning the set of features relevant to the Bayes classifier, which is slightly stronger than strong relevance as used here.

and write "predictor" to designate a variable used as input for a given classifier or regression model.

2. Background

We formalize the feature-selection task suited for our needs and provide relevant definitions. We do the same for the causal structure-learning task and prepare the needed basis for drawing the parallels between the two in the next section.

2.1. Feature Selection

We are given a data set of n samples $D = \{(\mathbf{x}_i, y_i),\ 1 \leq i \leq n\}$. Each data point $(\mathbf{x}_i, y_i)$ has $d-1$ inputs, modeled as a vector $\mathbf{x}_i \in \mathbb{R}^{d-1}$, and an output, or *target*, $y_i \in \mathbb{R}$ (we use $d-1$ and not d for the size of $\mathbf{x}_i$ for consistency with the rest of the paper). The data points are assumed to be drawn i.i.d. from a joint probability distribution over the random variables $\mathbf{V} = \mathbf{X} \cup \{Y\}$. The result of the feature-selection task we are interested in is a set of retained variables $\mathbf{F} \subseteq \mathbf{X}$. How many variables to retain and which variables to retain depends on the particular algorithm, and usually maximizes some tradeoff between efficiency and classification/regression error of a given learning task.

John et al. (1994) propose a classification of the input variables $\mathbf{X}$ with respect to their relevance to the target Y in terms of *conditional independence*.

Definition 1 (Conditional independence) *In a variable set* $\mathbf{V}$*, two random variables* X, Y *are* conditionally independent *given* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$*, noted* $(X \perp Y \mid \mathbf{Z})$*, if:*

$$\forall x, y, \mathbf{z} : P(X = x \mid Y = y, \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z}),$$

provided that $\forall \mathbf{z} : P(\mathbf{Z} = \mathbf{z}) > 0$*.*

Conditional independence is a generalization of the traditional notion of statistical independence. If two variables X and Y are independent, then the joint distribution is the product of the marginals: $P(X = x, Y = y) = P(X = x)P(Y = y)$. If they are dependent given some conditioning set $\mathbf{Z}$, then we can write $P(X = x, Y = y \mid \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z})P(Y = y \mid \mathbf{Z} = \mathbf{z})$. Conditional independence is a key concept in Bayesian networks (Pearl, 1988) because of the factorizations of the joint probability distribution it allows.

In feature selection, relevance of predictors to the target as proposed by John et al. (1994) is expressed in terms of conditional independence. In the following definitions, we write X_i to note the ith input variable, and $\mathbf{X}_{\setminus i}$ to note all input variables but the ith one.

Definition 2 (Strong relevance) *A variable* X_i *is* strongly relevant *to the target* Y *if*

$$P(Y \mid \mathbf{X}_{\setminus i}) \neq P(Y \mid \mathbf{X}_{\setminus i}, X_i).$$

A variable is strongly relevant to the target if it carries information about Y that no other variable carries. This is expressed in the definition by a change in the probability distribution of the target between conditioning on all other variables, $\mathbf{X}_{\setminus i}$, and also including X_i in the conditioning set. If X_i carries no exclusive information about Y, the two distributions will be identical.

Definition 3 (Weak relevance) *A variable X_i is* weakly relevant *to the target Y if it is not strongly relevant and*

$$\exists \mathbf{S} \subseteq \mathbf{X}_{\setminus i} : P(Y \mid \mathbf{S}) \neq P(Y \mid \mathbf{S}, X_i).$$

We speak of weak relevance of a variable X_i when there exists a certain context $\mathbf{S}$ in which it carries information about the target. However, this is not necessarily exclusive information, as it may be possible to obtain it from other variables.

Corollary 4 (Irrelevance) *A variable X_i is* irrelevant *to the target Y if it is neither strongly nor weakly relevant, that is, if*

$$\forall \mathbf{S} \subseteq \mathbf{X}_{\setminus i} : P(Y \mid \mathbf{S}) = P(Y \mid \mathbf{S}, X_i).$$

A variable is irrelevant if carries no information about the target at all, no matter what the context is.

For our purposes, we assume that the feature-selection algorithm returns the set of all strongly relevant variables, and only those.[5] (In Section 5, we discuss with experiments whether this is a reasonable assumption with the RFE algorithm.) Put in terms of conditional independence, the result $\mathbf{F}_Y$ of the feature-selection task with target Y is, with $\mathbf{V} = \mathbf{X} \cup \{Y\}$:

$$\mathbf{F}_Y = \left\{ X \;\middle|\; (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\}) \right\}. \tag{1}$$

That is the set of the variables that are dependent on the target Y, conditioned on all others. We need this property in Section 3 to use the output of the feature-selection task to build a causal graph. Note that if we repeat the feature-selection task using as target another variable $X \in \mathbf{V}$ yielding a result $\mathbf{F}_X$, we have:

$$X \in \mathbf{F}_Y \iff Y \in \mathbf{F}_X. \tag{2}$$

This follows as a direct consequence of (1) due to the symmetry of the conditional-independence relation $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ with respect to X and Y.

2.2. Causal Structure Learning

In causal structure learning, we are interested in representing graphically conditional dependencies found in the data. Under a set of assumptions, they have a causal interpretation. For this task, we have a data set of n samples $D = \{\mathbf{v}_i, 1 \leq i \leq n\}$. We do not designate a specific target variable in $\mathbf{V}$ as we are interested in learning the full structure of the network.

The graphical representation of choice for causal models is directed acyclic graphs (DAGs) (Pearl, 2000). In a causal graph represented by a DAG, we want to represent direct causal relations with arcs between pairs of variables. Choosing DAGs for this task implies restrictions, an obvious one of which is that causal feedback loops are excluded from the analysis. More formally, the joint probability distribution has to be *faithful* (or *DAG-isomorphic*, Pearl, 1988, p. 128); that is, there must exist a DAG that represents all (conditional) dependencies and independencies entailed by the distribution. Such a graph is called a *perfect map* of the distribution if there is a one-to-one mapping between the conditional-independence relation defined on variables and the *d-separation criterion* defined on the graphical nodes.

5. In the general case, this set can be empty without excluding the existence of other weakly relevant variables (Tsamardinos and Aliferis, 2003). In the next subsection, we detail the Faithfulness hypothesis, which allows us to exclude this particular case.

Definition 5 (*d*-separation) *In a DAG $\mathcal{G}$, two nodes X, Y are* d-separated *by $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, written $(X \nleftrightarrow Y \mid \mathbf{Z})$, if every path from X to Y is blocked by $\mathbf{Z}$. A path is* blocked *if at least one diverging or serially connected node is in $\mathbf{Z}$ or if at least one converging node and all its descendants are not in $\mathbf{Z}$. If X and Y are not* d*-separated by $\mathbf{Z}$, they are* d*-connected: $(X \leftrightarrow Y \mid \mathbf{Z})$.*

Determining whether two nodes in a graph are *d*-separated given some conditioning set is not visually immediate. It may for instance be unintuitive that whereas conditioning on a node W on a directed path $X \to W \to Y$ blocks the path from X to Y, conditioning on a common child Z of two variables X, Y in $X \to Z \leftarrow Y$ connects them. In the latter case, this common child is called a *collider*. If, furthermore, two parents X, Y of a node Z are nonadjacent in the full graph, then Z is called an *unshielded collider* for the pair (X, Y).

The definition of *d*-separation was worked out by Pearl (1988) to match as closely as possible the complicated nature of the conditional-independence relation with a graphical criterion, so that the class of faithful distributions, which can be represented by a perfect map, is as large as possible, while still keeping a natural graphical representation.

Definition 6 (Perfect map) *A DAG $\mathcal{G}$ is a* directed perfect map *of a joint probability distribution $P(\mathbf{V})$ if there is bijection between* d*-separation in $\mathcal{G}$ and conditional independence in P:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \nleftrightarrow Y \mid \mathbf{Z}) \iff (X \perp Y \mid \mathbf{Z}) \big). \tag{3}$$

If we take apart the perfect-map equivalence, we distinguish two important concepts, known as the Causal Markov condition and the Faithfulness condition (Spirtes et al., 2001, p. 29).

The **Causal Markov condition** is said to hold for a graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ and a probability distribution $P(\mathbf{V})$ if every variable is statistically independent of its graphical non-descendants (intuitively, of its non-effects, direct or indirect) conditional on its graphical parents (intuitively, its direct causes) in P. Pairs $\langle \mathcal{G}, P \rangle$ that satisfy the Causal Markov condition satisfy the implication

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \nleftrightarrow Y \mid \mathbf{Z}) \implies (X \perp Y \mid \mathbf{Z}) \big).$$

This is called *I-map property* by Pearl (1988).

The **Faithfulness condition** can be interpreted as the converse of the Causal Markov condition, and states that the only conditional independencies to hold are those specified by the Causal Markov condition:

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \leftrightarrow Y \mid \mathbf{Z}) \implies (X \not\perp Y \mid \mathbf{Z}) \big).$$

If the Causal Markov and Faithfulness conditions hold together for a pair $\langle \mathcal{G}, P \rangle$, then we find again the equivalence (3), and $\mathcal{G}$ is a perfect map of P.

In practice, the Causal Markov condition is used by the so-called constraint-based algorithms to perform conditional-independence tests on the data and build the graph accordingly, and Faithfulness is assumed to prove that the graph is correct. Hausman and Woodward (1999) discuss and explain in more detail the Causal Markov condition, and Steel (2005) discusses the Faithfulness condition and its motivations, pointing out cases where it can be violated. While the former is in general not violated simply by construction of the causal graph, violation of the latter occurs if the probability

distribution is not faithful. A simple example is the n-bit parity problem where the prior probability of each bit is uniform, of which the XOR problem is a special case: each variable is unconditionally independent of every other, but any variable pair becomes dependent conditioned on all other variables. On this problem, current constraint-based algorithms yield an empty graph because of the pairwise unconditional independencies, although it is not true that the data shows no dependency at all since one variable is a well-defined function of all others.

From this point on and for all proofs, we assume that the working data set D has a distribution that does not violate Faithfulness, and that it can thus be represented by a perfect map. In such a context, however, it is still not clear that causation can be inferred from conditional independence. We now proceed to explain the relation between causation and conditional independence.

Assuming Faithfulness, direct causation between X and Y, noted $X \twoheadrightarrow Y$, implies that X and Y are dependent given any conditioning set (Pearl and Verma, 1991, see definitions of potential and genuine causes):

$$X \twoheadrightarrow Y \implies \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big).$$

We denote the absence of direct causation by $X \not\twoheadrightarrow Y$. The exact converse of this implication does not hold. If we make the **Causal Sufficiency assumption** (Spirtes et al., 2001), that is, assume that no hidden common cause of two variables exists, we can write:

$$\big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big) \implies X \twoheadrightarrow Y \text{ or } Y \twoheadrightarrow X. \tag{4}$$

Using (4), we can theoretically determine all adjacencies of the causal graph with conditional-independence tests, but we cannot orient the edges. But there is a special causation pattern where conditional-independence tests can reveal the direction of causation. It is known as a **V-structure** (Pearl, 2000): two common causes X, Y, initially independent,[6] become dependent when conditioned on a common effect Z, then acting as a collider. This is noted $X \twoheadrightarrow Z \twoheadleftarrow Y$, where we also require $X \not\twoheadrightarrow Y$ and, symmetrically, $Y \not\twoheadrightarrow X$. Formally, we have:

$$\begin{aligned} &X \twoheadrightarrow Z \twoheadleftarrow Y \text{ and } X \not\twoheadrightarrow Y \text{ and } Y \not\twoheadrightarrow X \\ &\qquad \implies \big(\exists \mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y, Z\} : \big(X \perp\!\!\!\perp Y \mid \mathbf{S}\big) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big). \end{aligned}$$

The exact converse does not hold either. But using (4), we can find an equivalence relation defining a V-structure, still assuming Causal Sufficiency: first, we certify the existence of a link between X and Z and between Y and Z. Z is then identified as an unshielded collider if conditioning on it creates a dependency between X and Y:

$$\begin{aligned} X \twoheadrightarrow Z \twoheadleftarrow Y \iff \Big(&\big(\exists \mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y, Z\} : \big(X \perp\!\!\!\perp Y \mid \mathbf{S}\big) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big) \\ &\text{and } \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X, Z\} : (X \not\perp\!\!\!\perp Z \mid \mathbf{S})\big) \\ &\text{and } \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{Y, Z\} : (Y \not\perp\!\!\!\perp Z \mid \mathbf{S})\big)\Big). \end{aligned} \tag{5}$$

6. The two causes X and Y actually do not need to be unconditionally independent, but there must exist a (possibly empty) separating set $\mathbf{S}_{XY} \subseteq \mathbf{V} \setminus \{X, Y\}$ such that $\big(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY}\big)$ for the collider to be identifiable. This implies that no direct causation $X \twoheadrightarrow Y$ or $Y \twoheadrightarrow X$ may exist: the collider must be unshielded.

Actually, typical algorithms first establish the existence of a link between two variables by seeking a certificate equivalent to, or implicating the premise of, (4), and then look for orientation possibilities. Note that there is no guarantee that all links can be oriented into causal arcs, and that in general we therefore cannot recover the full causal structure with conditional-independence tests. This is the problem known as **causal underdetermination** (Spirtes et al., 2001, p. 62): for the structure-learning task given observational data, a correct graph is specified by its adjacencies and its V-structures only. Partially oriented graphs returned by structure-learning algorithms represent *observationally equivalent classes* of causal graphs (Pearl, 2000, p. 19). This means that for a given joint probability distribution $P(\mathbf{V})$, the set of all conditional-independence statements that hold in P does not yield a unique perfect map in general.

Formally, if we combine (3), (4) and (5), we find, for a perfect causal map $\mathcal{G}$ (using the symbol "$\twoheadrightarrow$" to denote direct causation and "$\rightarrow$" to denote an arc in the graph):

$$\begin{aligned} X, Y \text{ adjacent in } \mathcal{G} &\iff X \twoheadrightarrow Y \text{ or } Y \twoheadrightarrow X \\ X \rightarrow Z \leftarrow Y &\iff X \twoheadrightarrow Z \twoheadleftarrow Y. \end{aligned} \tag{6}$$

It is sometimes possible to orient further arcs in a graph by looking at already-oriented arcs and propagating constraints, preventing acyclicity and the creation of additional V-structures other than those already detected. The graph after this constraint-propagation step is called *completed PDAG*, *maximally oriented PDAG* (CPDAG), or *essential graph*, depending on the author.

3. Causal Network Construction Based on Feature Selection

We have looked at the ideal outcome of feature selection in (1) and how to read a causal graph in (6). We now turn to showing how feature selection can be used to build a causal graph. From now on and for the rest of this paper, we assume that the joint probability distribution over all variables $\mathbf{V}$ is faithful.

3.1. Identifying the Markov Blankets

In the context of directed graphical models, the Markov blanket of a node X, noted $\mathbf{Mb}(X)$, is the set of parents, children, and children's parents (spouses) of X. As an easy property, note that we have:

$$X \in \mathbf{Mb}(Y) \iff Y \in \mathbf{Mb}(X).$$

The following statement is a key property of Markov blankets.

Property 7 (Total conditioning) *In the context of a faithful causal graph $\mathcal{G}$, we have:*

$$\forall X, Y \in \mathbf{V}: \ \big(X \in \mathbf{Mb}(Y) \iff (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\})\big).$$

(See Appendix A for the proof.) This property says that the Markov blanket of each node is the set of all variables that are dependent on it, conditioned on all other variables. In other words, in a causal graph, the parents, children, and spouses of Y store information about Y that cannot be obtained from any other variable. Note that $\mathbf{Mb}(Y)$ then has exactly the property of the output of feature selection, $\mathbf{F}_Y$, as characterized in (1). This links feature selection and causal structure learning in the sense that

$$\mathbf{F}_Y = \mathbf{Mb}(Y),$$

the Faithfulness assumption guaranteeing the unicity of $\mathbf{Mb}(Y)$. However, Markov blankets alone do not fully specify a causal graph. Thus, feature selection, even if guaranteed to find only strongly relevant features, cannot be directly used to construct the graph as we want it to be. The problem is that spouses of Y, even if not directly linked in the original graph, would be linked in $\mathbf{F}_Y$ and $\mathbf{Mb}(Y)$. An additional step is needed to transform the Markov blankets into parents, children, and spouses.

3.2. Recovering the Local Structure

The result of feature selection can be graphically shown by an undirected graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ where $(X, Y) \in \mathbf{E} \Leftrightarrow X \in \mathbf{F}_Y$. This graph is close to the original causal graph in that it contains all arcs as undirected links, and additionally links spouses together, and is called the *moral graph* of the original directed graph (Lauritzen and Spiegelhalter, 1988, p. 166). The extra step needed to transform this graph into a causal PDAG is the deletion of the spouse links and the orientation of the arcs, a task which we call "resolving the Markov blankets."

An existing algorithm can resolve the Markov blankets, that is, use Markov blanket information to infer the local structure around a node: the Grow-Shrink (GS) algorithm, proposed by Margaritis and Thrun (1999). The full algorithm first finds the Markov blanket for each variable, and performs further conditional-independence tests around each variable to infer the structure locally. It then uses a heuristics to remove cycles possibly introduced by previous steps. We list in Algorithm 1 (using our notation) the steps of the algorithm responsible for building the local structure using the Markov blanket information, as this is exactly the task we are trying to solve. In the code, $\mathbf{Bd}(X)$ stands for the *boundary* of X; that is, the set of its direct neighbors in the graph $\mathcal{G}$. It is different from $\mathbf{Mb}(X)$ in that whereas $\mathbf{Mb}(X)$ is passed as input to the algorithm and is fixed, $\mathbf{Bd}(X)$ depends on the graph $\mathcal{G}$, which is modified throughout the algorithm. We note a conditional-independence test with a subroutine call to $\textsc{CondIndep}(X, Y, \mathbf{Z})$: ideally, this function returns *true* when $(X \perp Y \mid \mathbf{Z})$ holds, and *false* otherwise. More will be said about the actual implementation of such tests in Section 4. The command **break** is used to break out of the innermost loop, saving unnecessary computations.

The GS algorithm makes two passes over all variables and the members of their Markov blankets (or direct neighbors in the second pass). It first removes the possible spouse links between linked variables X and Y by looking for a d-separating set around X and Y. In a second pass, it orients the arcs whenever it finds that conditioning on a middle node creates a dependency. While searching for the appropriate conditioning set, GS selects the smallest base search set (set $\mathbf{B}$ in Algorithm 1) for each phase. This has two very desirable effects. First, it reduces the number of tests, which is useful because each phase contains a subset search, exponential in time complexity with respect to the searched set. Second, it reduces the average size of the conditioning set, which increases the power of the statistical tests, and thus helps reduce the number of Type II errors.

While the GS approach considerably reduces the number of tests to be performed with respect to a large subset search, it is possible to perform fewer tests while still identifying correctly the structure and orienting the arcs, and decreasing the average conditioning set size. A helpful observation is that orientation and removal of the spouse links can be done together in a single pass. We know, as discussed in the previous section, that only arcs in V-structures can be oriented: fortunately, V-structures are exactly spotted when we identify a spouse link to be removed. Two spouses X and Y that are not directly linked in the original causal graph can be d-separated by some set

Algorithm 1: Resolve the Markov Blankets with the Grow-Shrink Algorithm

procedure RESOLVEMARKOVBLANKS_GROWSHRINK
Input: $\mathbf{Mb}(\cdot)$: the Markov blanket information for each node $X \in \mathbf{V}$
Output: $\mathcal{G}$: partially oriented DAG

`/* Compute graph structure */`
$\mathcal{G} \leftarrow$ moral graph according to $\mathbf{Mb}(\cdot)$
foreach $X \in \mathbf{V}$ **and** $Y \in \mathbf{Mb}(X)$ **do**
  $\mathbf{B} \leftarrow$ smallest set of $\{\mathbf{Bd}(X) \setminus \{Y\},\ \mathbf{Bd}(Y) \setminus \{X\}\}$
  foreach $\mathbf{S} \subseteq \mathbf{B}$ **do**
    if CONDINDEP$(X, Y, \mathbf{S})$ **then** remove link $X - Y$ from $\mathcal{G}$; **break**
  end
end
`/* Orient edges */`
foreach $X \in \mathbf{V}$ **and** $Y \in \mathbf{Bd}(X)$ **do**
  foreach $Z \in \mathbf{Bd}(X) \setminus \mathbf{Bd}(Y) \setminus \{Y\}$ **do**
    orient $Y \rightarrow X$ `/* to be corrected if a test yields independence */`
    $\mathbf{B} \leftarrow$ smallest set of $\{\mathbf{Mb}(Y) \setminus \{Z\},\ \mathbf{Mb}(Z) \setminus \{Y\}\}$
    foreach $\mathbf{S} \subseteq \mathbf{B}$ **do**
      if CONDINDEP$(Y, Z, \mathbf{S} \cup \{X\})$ **then**
        remove orientation $Y \rightarrow X$; **break**
      end
    end
    if $Y \rightarrow X$ **then break**
  end
end
return $\mathcal{G}$
end

of nodes. Thus, if we can find a set $\mathbf{S}_{XY}$ that makes X and Y conditionally independent, we know that the link between them is a spouse link to be removed. Moreover, we know that any node Z part of the intersection of their Markov blankets not included in $\mathbf{S}_{XY}$ is a collider and thus a common child, and that the triplet (X, Z, Y) is actually a V-structure $X \rightarrow Z \leftarrow Y$ in the original graph. This follows from the definition of d-separation. What we need is an efficient search algorithm to find such d-separating sets.

An approach based on this observation has two main benefits. First, it only searches the triangles, that is, the cliques of three nodes, in the moral graph. Assuming that the information about the Markov blanket is correct, only triangles can hide spouse links and V-structures. Second, for each connected pair $X - Y$ in a triangle, decisions about possible spouse links and arc orientation are taken together and thus faster.

Pseudocode for the proposed search algorithm is listed in Algorithm 2, where the notation $\mathcal{G}^{\backslash XY}$ denotes the moral graph $\mathcal{G}$ where all direct links involving X or Y have been removed. The algorithm uses the following concept.

Definition 8 (Collider sets) *In an undirected graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$, let $\mathbf{Tri}(X - Y)$ (with $X, Y \in \mathbf{V}$ and $(X, Y) \in \mathbf{E}$) be the set of vertices forming a triangle with X and Y:*

$$\mathbf{Tri}(X - Y) = \{Z \in \mathbf{V} \mid (X, Z) \in \mathbf{E}, (Y, Z) \in \mathbf{E}\}.$$

Suppose that $\mathcal{G}$ is the moral graph of the DAG representing the causal structure of a faithful data set. A set of vertices $\mathbf{Z} \subseteq \mathbf{Tri}(X - Y)$ then has the Collider Set *property for the pair (X, Y) if it is the largest set that fulfills*

$$\exists \mathbf{S}_{XY} \subseteq \mathbf{V} \setminus \{X, Y\} \setminus \mathbf{Z} : (X \perp Y \mid \mathbf{S}_{XY}) \tag{7}$$

$$\text{and} \quad \forall Z_i \in \mathbf{Z} : (X \not\perp Y \mid \mathbf{S}_{XY} \cup \{Z_i\}). \tag{8}$$

The set $\mathbf{S}_{XY}$ is then a d*-separating set for X, Y.*

Lemma 9 *In the context of a faithful causal graph, the set $\mathbf{Z}$ that has the Collider Set property for a given pair (X, Y) exists if and only if X is neither a direct cause nor a direct effect of Y. This set $\mathbf{Z}$ is unique when it exists. (Proof in Appendix A.)*

The purpose of Algorithm 2 is thus to find these collider sets (in the pseudocode, the symbol $\subsetneq$ denotes the strict subset relation). The algorithm loops over all triangle links and performs a collider set search for each of them. Let $X - Y$ be one of these links: if it is not a spouse link, the search procedure will leave the value of the d-separating set $\mathbf{S}_{XY}$ to its default value, **null**. Otherwise, $\mathbf{S}_{XY}$ will be set to a (possibly empty[7]) set for X and Y. The collider set can be inferred by removing the d-separating set from the triangle nodes $\mathbf{Tri}(X - Y)$: as $\mathbf{Tri}(X - Y)$ contains nodes on a path of length 2 between X and Y, finding a d-separating set that does not contain some of these nodes proves that they can only be colliders according to the definition of d-separation.[8] For instance, if the procedure produces an empty set for a given linked pair $X - Y$, then X and Y are unconditionally independent, and therefore all nodes in $\mathbf{Tri}(X - Y)$ are colliders.

Two caveats have to be observed during this search, however. First, there might be other active, d-connecting paths between X and Y that are not going through any

7. Note that returning an empty d-separating set in $\mathbf{S}_{XY}$ is different from returning **null**, signaling the absence of any such set.
8. The next paragraphs describe patterns where this is not true and show how the algorithm still deals with them correctly.

Algorithm 2: Resolve the Markov Blankets with Collider Sets

procedure RESOLVEMARKOVBLANKETS_COLLIDERSETS
Input: $\mathbf{Mb}(\cdot)$: the Markov blanket information for each node $X \in \mathbf{V}$
Output: $\mathcal{G}$: partially oriented DAG

1 $\mathcal{G} \leftarrow$ moral graph according to $\mathbf{Mb}(\cdot)$
2 $\mathbf{C} \leftarrow \{\}$, an empty list of orientation directives
3 **foreach** edge $X - Y$ part of a fully connected triangle **do**
4 $\quad \mathbf{S}_{XY} \leftarrow$ **null** /* search for d-separating set */
5 $\quad \mathbf{B} \leftarrow$ smallest set of $\{\mathbf{Bd}(X) \setminus \mathbf{Tri}(X-Y) \setminus \{Y\},\ \mathbf{Bd}(Y) \setminus \mathbf{Tri}(X-Y) \setminus \{X\}\}$
6 $\quad$ **foreach** $\mathbf{S} \subsetneq \mathbf{Tri}(X-Y)$ /* subset search */
7 $\quad$ **do**
8 $\quad\quad \mathbf{Z} \leftarrow \mathbf{B} \cup \mathbf{S}$
9 $\quad\quad$ **if** CONDINDEP$(X, Y, \mathbf{Z})$ **then**
10 $\quad\quad\quad \mathbf{S}_{XY} \leftarrow \mathbf{Z}$
11 $\quad\quad\quad$ **break** to line 24
12 $\quad\quad$ **end**
13 $\quad\quad \mathbf{D} \leftarrow \mathbf{B} \cap \left\{\text{nodes reachable by } W \text{ in } \mathcal{G}^{\setminus XY} \mid W \in \left(\mathbf{Tri}(X-Y) \setminus \mathbf{S}\right)\right\}$
14 $\quad\quad \mathbf{B}' \leftarrow \mathbf{B} \setminus \mathbf{D}$
15 $\quad\quad$ **foreach** $\mathbf{S}' \subsetneq \mathbf{D}$ /* descendant of collider may be opening a path */
16 $\quad\quad$ **do**
17 $\quad\quad\quad \mathbf{Z} \leftarrow \mathbf{B}' \cup \mathbf{S}' \cup \mathbf{S}$
18 $\quad\quad\quad$ **if** CONDINDEP$(X, Y, \mathbf{Z})$ **then**
19 $\quad\quad\quad\quad \mathbf{S}_{XY} \leftarrow \mathbf{Z}$
20 $\quad\quad\quad\quad$ **break** to line 24
21 $\quad\quad\quad$ **end**
22 $\quad\quad$ **end**
23 $\quad$ **end**
24 $\quad$ **if** $\mathbf{S}_{XY} \neq$ **null** /* save orientation directive */
25 $\quad$ **then**
26 $\quad\quad$ mark link $X - Y$ as spouse link in $\mathcal{G}$
27 $\quad\quad$ **foreach** $Z \in \left(\mathbf{Tri}(X-Y) \setminus \mathbf{S}_{XY}\right)$ **do**
28 $\quad\quad\quad \mathbf{C} \leftarrow \mathbf{C} \cup \{(X \rightarrow Z \leftarrow Y)\}$
29 $\quad\quad$ **end**
30 $\quad$ **end**
31 **end**
32 remove all spouse links (i.e., marked links) from $\mathcal{G}$
33 **foreach** orientation directive $(X \rightarrow Z \leftarrow Y) \in \mathbf{C}$ /* orient edges */
34 **do**
35 $\quad$ **if** edges $X - Z$ and $Y - Z$ still exist in $\mathcal{G}$ **then**
36 $\quad\quad$ orient edges as $X \rightarrow Z \leftarrow Y$
37 $\quad$ **end**
38 **end**
39 **return** $\mathcal{G}$
end

node of $\mathbf{Tri}(X - Y)$. Those nodes must be blocked by appropriate conditioning on the boundary of X or Y as determined by the base conditioning set at line 5. Second, this base conditioning set must be checked not to include any descendant of possible colliders. If it did, it would open a d-connecting path according to Definition 5. This check is performed at lines 13 to 21. At line 13, we build a set $\mathbf{D}$ that includes all possible descendants of currently conjectured colliders that intersect our base conditioning set $\mathbf{B}$. The following loop makes sure none of them was opening a path between X and Y.

Theorem 10 *In the large sample limit, for faithful, causally sufficient data sets, the procedure* RESOLVEMARKOVBLANKETS_COLLIDERSETS *correctly identifies all V-structures and all spouse links, assuming consistent statistical tests. (Proof in Appendix A.)*

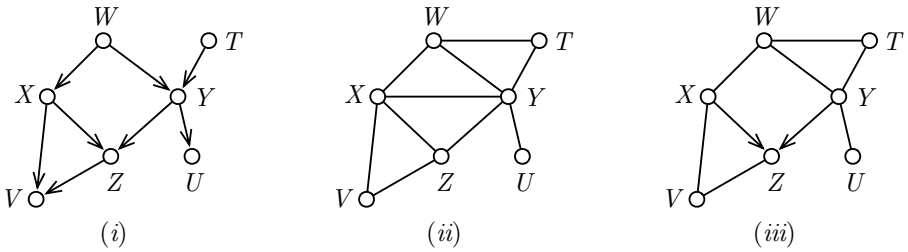


Figure 1: Sample local causal structure (i) and corresponding moral graph (ii). On (iii), the spouse link and orientation information that the collider set search for the linked pair $X - Y$ gives.

This procedure is best understood with a graphical example. Consider the sample local structure in Figure 1, imagine it is part of a larger network, and suppose we are performing the search starting at line 4 in Algorithm 2. We are looking for a d-separating set for X and Y. Looking at the original graph, we see that $\{W\}$ is the smallest such set; let us see how the algorithm finds it. We have: $\mathbf{Tri}(X - Y) = \{W, Z\}$, $\mathbf{Bd}(X) = \{W, Y, Z, V\}$ and $\mathbf{Bd}(Y) = \{W, X, Z, U, T\}$. The base conditioning set $\mathbf{B}$ will thus be the smallest of $\big\{\{V\}, \{U, T\}\big\}$, thus $\mathbf{B} = \{V\}$. At this stage, conditioning on V is justifiable: one cannot exclude situations where X and Y are d-connected given the empty set through T and V, for instance if T and V both had a common cause farther away in the network. But actually in this example, all (perfect) tests containing V in the conditioning set will yield dependence, because it is a descendant of the collider Z and thus opens a path by definition of d-separation. Eventually, in the iteration where $\mathbf{S} = \{W\}$, we will find conditional independence in the nested loop at lines 15 to 21. As $\mathbf{Tri}(X - Y) \setminus \mathbf{S} = \{Z\}$, $\mathbf{D}$ will be assigned the value $\{V\}$ and $\mathbf{B}'$ will be empty, so that we will perform exactly one extra test at line 18 with the conditioning set $\mathbf{S}_{XY} = \{W\}$, which yields independence. This in turn allows us to identify the link $X - Y$ as a spouse link and determine (line 27) that the set $\mathbf{Tri}(X - Y) \setminus \mathbf{S}_{XY} = \{Z\}$ is the set of all direct effects of X and Y; that is, fulfills the Collider Set property.

For some structures, the order in which arcs are removed and oriented must happen such that all spouse links are removed before proceeding to orientation. Consider another example, shown in Figure 2, and suppose again that that we are looking for a

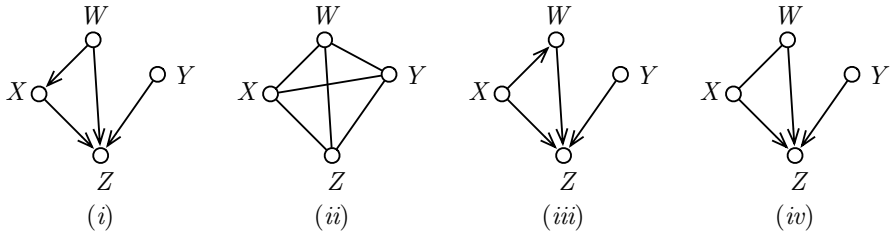


Figure 2: Another sample local causal structure (i) and corresponding moral graph (ii). On (iii), a wrong result if orientation is done immediately at line 28 of Algorithm 2. On (iv), the correct (non-)orientation if the condition at line 35 is added.

d-separating set for the pair (X, Y). As X and Y are unconditionally independent, $\mathbf{S}_{XY} = \emptyset$ is a valid d-separating set. We may thus remove the link $X - Y$, and considering that $\mathbf{Tri}(X - Y) = \{W, Z\}$, we could want to orient $X \rightarrow Z \leftarrow X$ and $X \rightarrow W \leftarrow X$ (leaving the spouse link $W - Y$ to be removed later). This would be wrong, precisely because $W - Y$ is a spouse link, and thus the orientation $X \rightarrow W \leftarrow X$ is not allowed if one of the links to be oriented does not actually exist in the original graph. This is the reason why all orientation directives are saved in a list $\mathbf{C}$ at line 28 of Algorithm 2. After all spouse links have been removed, the orientations are done at line 36 only when both links to be oriented still exist, thus ensuring the existence of the V-structure $X \rightarrow Z \leftarrow Y$.

We do not claim that our algorithm uses the smallest possible conditioning set for the tests. There is a tradeoff between obtaining the minimal possible conditioning set and keeping the total number of tests low in the average case. In the empirical evaluation of this algorithm, we examine three behavioral criteria: the total number of tests, the average size of the conditioning set, and the maximum size of the conditioning set.

The complexity of the whole algorithm iterating over all triangle links, in terms of number of calls to CONDINDEP, is $\mathcal{O}(d^2 2^\alpha)$, where d is the number of variables and $\alpha = \max_{X \in \mathbf{V}} |\mathbf{Mb}(X)| - 1$. In the worst case of a fully connected graph, where $\mathbf{Mb}(X) = \mathbf{V} \setminus \{Y\}$, it is exponential in the number of variables due to the subset search. But in practice, the original graphs are often sparse enough so that the actual run time is not exponential. Many algorithms (e.g., MMMB, HITON_MB, AlgorithmMB, GS) perform subset searches in the (possibly augmented) Markov blanket and thus rely on graph sparseness to be efficient. Although the complexity of RESOLVEMARKOVBLANKETS_COLLIDERSETS is the same as that of RESOLVEMARKOVBLANKETS_GROWSHRINK, we show in the experimental results in Section 5 that the former performs fewer tests with a smaller average conditioning set size, while still providing comparable accuracy in structure learning.

3.3. A Generic Algorithm Based on Feature Selection

Thanks to the subroutine explained in the previous section, we can now draft a generic algorithm for structure learning based on feature-selection methods returning strongly relevant features. Algorithm 3 lists pseudocode for the three main steps of this approach:

1. Find the conjectured Markov blanket of each variable with feature selection and build the moral graph;
2. Remove spouse links and orient V-structures using collider sets;
3. Propagate orientation constraints.

For the sake of completeness, the constraint propagation rules of Step 3 have also been listed, in a separate subroutine (see Algorithm 4). They are common in structure learning to obtain a completed PDAG (Pearl and Verma, 1991). Meek (1995) proves that these three rules indeed return the maximally oriented graph when given a PDAG whose V-structures are oriented.

Algorithm 3: Causal Structure Learning with Feature Selection

procedure GENERICSTRUCTURELEARNING
Input: D: $n \times d$ data set with n d-dimensional data points
Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

`/* Step 1: Markov blanket construction */`
1 **foreach** variable $X \in \mathbf{V}$ **do**
2 $\quad \mathbf{F}_X \leftarrow$ FEATURESELECTIONALGORITHM(X, D)
3 **end**
4 **foreach** pair (X, Y) such that $Y \in \mathbf{F}_X$ and $X \in \mathbf{F}_Y$ `/* symmetry check */`
5 **do**
6 $\quad$ add X to $\mathbf{Mb}(Y)$ and Y to $\mathbf{Mb}(X)$
7 **end**

`/* Step 2: Spurious arc removal & V-structure detection */`
8 $\mathcal{G} \leftarrow$ RESOLVEMARKOVBLANKETS$(\mathbf{Mb}(\cdot))$

`/* Step 3: Constraint propagation */`
9 $\mathcal{G} \leftarrow$ COMPLETEPDAG$(\mathcal{G})$
10 **return** $\mathcal{G}$
end

The challenge with this approach is twofold. One issue is efficiency: consistent but slow feature-selection algorithms will not beat existing causal learning algorithms, as they have to be run as many times as the number of variables d. The second and biggest issue is that consistent feature-selection algorithms are needed in order to prove correctness of this generic algorithm, in the sense that the result of feature selection should be equal to the set of strongly relevant features. This requirement is not always fulfilled. Hardin et al. (2004) study an SVM classifier and discuss feature selection based on the $\mathbf{w}$ weights: although irrelevant variables are not selected in the large sample limit, they show that the weights of the weakly relevant variables can be as close as one wishes to that of the strongly relevant variables due to the large-margin behavior of SVMs. Forward feature selection has been shown to miss strongly relevant variables (Guyon and Elisseeff, 2003). Nilsson et al. (2007) also describe forward selection as inconsistent, but claim that backward feature elimination is actually consistent in the large-sample limit.[9] For finite data sets, Statnikov et al. (2006) further show (among others) that even the weights of the irrelevant variables can get bigger than that of

9. This is subject to the assumption that the underlying classifier must itself be consistent, in the sense that it must return the Bayes classifier in the large-sample limit.

Algorithm 4: Orient a PDAG maximally

procedure COMPLETEPDAG
Input: $\mathcal{G}$: partially directed acyclic graph
Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

1 **while** $\mathcal{G}$ is changed by some rule /* fixed-point iteration */
2 **do**
3 **foreach** X, Y, Z such that $X \to Y - Z$ **do**
4 orient as $X \to Y \to Z$ /* no new V-structure */
5 **end**
6 **foreach** X, Y such that $X - Y$ and $\exists$ directed path from X to Y **do**
7 orient as $X \to Y$ /* preserve acyclicity */
8 **end**
9 **foreach** X, Y s.t. $X - Y$ & $\exists$nonadj. Z, W s.t. $X - Z \to Y$ & $X - W \to Y$ **do**
10 orient as $X \to Y$ /* three-fork V with married parents */
11 **end**
12 **end**
13 **return** $\mathcal{G}$
end

relevant variables, and that weakly relevant variables can be selected more often than strongly relevant variables in some cases.

These considerations are taken into account in our approach. In the next section, we describe an instantiation of the generic algorithm with an existing backward feature-elimination algorithm. Expecting the feature selection to be too inclusive, that is, to include features that are not strongly relevant, we add the filtering condition at line 4 of the generic outline in Algorithm 3: in order to link X and Y in the moral graph, we require the feature selection performed for X to have selected variable Y, and conversely, we require X to have been selected by the feature selection performed for Y. This does not theoretically guarantee the absence of "false positives," however. Further in the section, we replace the feature-selection step with a provably consistent algorithm in the multivariate Gaussian case, and analyze its complexity and behavior.

4. Algorithms for Causal Feature Selection

In this section, we show two algorithms (and a variant) as instantiations of the generic approach previously described. First, we explain an algorithm based on the Recursive Feature Elimination (RFE) algorithm (Guyon et al., 2002) as a direct application of existing methods. We then describe Total Conditioning (TC), a fast algorithm that can be proved correct under the multivariate Gaussian assumption. We also show a variant, TC_{bw}, that improves accuracy with low sample sizes by using an explicit backward feature-selection heuristics. In Section 5, we report on experiments including these algorithms.

4.1. An RFE-Based Approach

To empirically test the soundness of the approach, we propose to use RFE over a Support Vector Regression (SVR) learner (Smola and Schölkopf, 1998) with a linear

kernel, assuming for this example that we will deal with multivariate Gaussian data. RFE is an instance of a backward feature-elimination algorithm. Given some learner (in this case, SVR), it iteratively trains it, ranks the features according to some criterion, and remove the feature (or the p features) with the smallest ranking criterion. This criterion can be the weights $\mathbf{w}$ attributed to the features by the learner, or some sensitivity measure of the features (Guyon et al., 2002). In our case, we used the weights $\mathbf{w}$ of SVR as described in Smola and Schölkopf (1998).

Using RFE, the Markov blanket identification is done in two steps:

1. Use RFE to rank the predictors according to their weights in the trained model and to provide what can be seen as a relevance ordering of the predictors;

2. Determine the size of the Markov blanket and thus the number of variables to select from the list returned by RFE.

We do not have a theoretical guarantee that RFE/SVR will return the Markov blanket variables. Although Nilsson et al. (2007) shows that RFE/SVM as described in Guyon et al. (2002) is consistent (i.e., returns strongly relevant variables in the large-sample limit), the limitations of ranking variables on the $\mathbf{w}$ weights of an SVM with finite data sets have also been highlighted (Hardin et al., 2004; Statnikov et al., 2006). For now, we thus use this feature-selection step as a heuristics.

In order to determine the number of variables to select from the ranked list returned by RFE, we use the following criterion: starting with the first variable from the list, accept a new variable in the Markov blanket if the cross-validated training error of the SVR decreases with the new variable, and stop and return the current list if adding the next variable increases the error.

Algorithm 5: An RFE-Based Feature-Selection Step

procedure RFEFEATURESELECTION
  Input: X: the target variable to perform feature selection for
    D: $n \times d$ data set with n d-dimensional data points
  Output: $\mathbf{S}$: the set of selected variables

1 $\mathbf{w} \leftarrow$ weights of $\mathbf{V} \setminus X$ according to RFE(SVR)
2 $\mathbf{P} \leftarrow$ predictor variables sorted according to $\mathbf{w}$
3 $\mathbf{S} \leftarrow \emptyset$
4 $error_{opt} \leftarrow \mathrm{var}[X]$ /* MSE of constant function */
5 $error \leftarrow$ TRAIN(cross-validated SVR with predictor $(\mathbf{P})_1$))
6 **while** $error < error_{opt}$ **do**
7 $error_{opt} \leftarrow error$
8 $\mathbf{S} \leftarrow \mathbf{S} \cup \{ (\mathbf{P})_1 \}$ /* add beneficial predictor */
9 $\mathbf{P} \leftarrow \mathbf{P} \setminus \{ (\mathbf{P})_1 \}$
10 $error \leftarrow$ TRAIN(cross-validated SVR with predictors $\mathbf{S} \cup \{ (\mathbf{P})_1 \}$)
11 **end**
12 **return S**
end

The symmetry condition (2), $X \in \mathbf{F}_Y \Leftrightarrow Y \in \mathbf{F}_X$, might not be satisfied: we rely on the check at line 4 of the generic approach of Algorithm 3 to make sure that we do not select spurious features in the Markov blanket. This conservative approach implies that

we expect RFE to select at least all strongly relevant variables, plus possibly some others that we hope to identify with this simple condition.

As a conditional-independence test at lines 9 and 18 of the collider set search in Algorithm 2, we can use the distribution-free Recursive Median (RM) algorithm proposed by Margaritis (2005) to detect the V-structure and remove the spouse links, or a z-test as used in Scheines et al. (1995) in the case of Gaussian data.

Although we expect the resulting graph to be accurate in the large sample limit (see Section 5), we also expect the run time of such an approach to be much higher compared to existing algorithms. Training the SVR has a cubic complexity in terms of the number of samples, $\mathcal{O}(n^3)$. To get an accurate ranking, RFE runs the training $d-1$ times. Then, a new SVR learner is trained and cross-validated several times (we used a 5-fold cross-validation) to get the validation error, which is repeated for each variable in the actual Markov blanket. The complexity for the whole feature-selection step is then $\mathcal{O}(d^2n^3)$, with a large constant factor. We thus emphasize that this RFE-based feature selection is not meant as a valid practical instantiation of the generic algorithm, but rather as a proof of concept to validate the approach. In order to be practical, the feature-selection step has to be redesigned so that it is done efficiently when run for all variables. This is what the next algorithm is meant to address in the specific case of multivariate Gaussian variables.

4.2. The TC Algorithm

We now propose in the procedure TCFEATURESELECTION (Algorithm 6) another instantiation of the feature-selection call at line 2 of the generic approach of Algorithm 3. The whole algorithm as determined by the feature-selection, collider-identification, and maximal-orientation steps is equivalent to the TC algorithm described in Pellet and Elisseeff (2007). (We thus write "TC" to refer to the whole algorithm and not only to the feature-selection procedure, referred to as TCFEATURESELECTION.)

For a given target variable X, TC estimates the coefficients of a multiple regression problem, considering all other variables $\mathbf{V} \setminus X$ as predictors. It then returns the significant predictors, according to a t-test on the coefficient of each variable. Its short listing is in Algorithm 6.

Algorithm 6: The Total Conditioning Feature-Selection Step

procedure TCFEATURESELECTION
  X: the target variable to perform feature selection for
  Input: D: $n \times d$ data set with n d-dimensional data points
  Output: **S**: the set of selected variables
1 **b** $\leftarrow$ weights of $\mathbf{V} \setminus X$ in the problem of regressing X on $\mathbf{V} \setminus X$
2 **S** $\leftarrow$ {predictors whose b weight is significant}
3 **return S**
end

The conditional-independence tests to be performed at lines 9 and 18 of the collider set search of Algorithm 2 are done using partial correlation.

Definition 11 (Partial correlation) *In a variable set* $\mathbf{V}$*, the* partial correlation *between two random variables* $X, Y \in \mathbf{V}$ *given* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$*, noted* $\rho_{XY \cdot \mathbf{Z}}$*, is the correlation of the*

residuals R_X and R_Y resulting from the least-squares linear regression of X on $\mathbf{Z}$ and of Y on $\mathbf{Z}$, respectively.

TC was shown to be correct in the large sample limit (subject to the consistency of the statistical tests) in Pellet and Elisseeff (2007) under the Faithfulness and Causal Sufficiency assumptions. For the sake of completeness, we add the proof to Appendix A. The main points leading to the correctness of TC are the equivalence of a zero regression weight for some predictor Y while regressing X on all variables $\mathbf{V} \setminus X$ and a zero partial correlation $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}}$, and the fact that this is zero if and only if $\left(X \perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\}\right)$ holds in a Gaussian context (Baba et al., 2004). Then, our feature-selection step (Algorithm 6) gives the Markov blanket for each node, and the collider set search (Algorithm 2) then takes care of identifying the V-structures and removing the spouse links.

The other advantage of using linear regression and partial correlation is that it yields a fast algorithm. Actually, *all* regression weights and parameters needed for the feature-selection step of TC can be efficiently computed by inverting the sample correlation matrix $\mathbf{R} \in [-1, 1]^{d \times d}$. Building graphs by inverting the correlation matrix is typically what is done with Gaussian Markov random fields, a special case of undirected graphical models (see, e.g., Talih, 2003).

The weight computation and the statistical significance tests are performed as follows. Let $\hat{b}_{ij}$ be the maximum likelihood estimator of the true regression weight b_{ij} of predictor X_j when X_i is the dependent variable, such that it solves the multiple regression equation for target X_i in the sense that it minimizes the sum of the squared residuals

$$SS_R = \sum_{k=1}^{n} \left(x_{ik} - \sum_{j=1, j \neq i}^{d} \hat{b}_{ij} x_{jk} \right)^2$$

where x_{ik} is the value of X_i for the kth sample. If we have the inverse correlation matrix $\mathbf{R}^{-1} = (r^{ij})$, the vector $\mathbf{b}$ at line 1 of Algorithm 7 can be found in linear time: $\hat{b}_{ij} = -r^{ij}/r^{ii}$ (Raveh, 1985). For instance, the list of weights to predict variable X_1 with all others is

$$\mathbf{b}_1 = (\hat{b}_{12}, \hat{b}_{13}, \cdots, \hat{b}_{1d}) = -(r^{12}, r^{13}, \cdots, r^{1d})/r^{11}. \tag{9}$$

The distribution of these weights is known (Judge et al., 1988):

$$\frac{\hat{b}_{ij} - b_{ij}}{\hat{\sigma}_{ij}} \sim t_{(n-(d-1))}, \tag{10}$$

where $\hat{\sigma}_{ij}$ is the standard error of the jth predictor for variable X_i; that is, that it follows a t distribution with a number of degrees of freedom df = number of samples − number of predictors $= n - (d - 1)$. For our null hypothesis $H_0 : b_{ij} = 0$, we need $\hat{\sigma}_{ij}$ in addition to $\hat{b}_{ij}$ to compute the t-statistics $\hat{b}_{ij}/\hat{\sigma}_{ij}$. The estimate of the coefficient error $\hat{\sigma}_{ij}$ can be expressed as

$$\hat{\sigma}_{ij} = \hat{\sigma}_i \sqrt{\omega^{jj}/n},$$

where $\hat{\sigma}_i$ is an estimator of the standard error of the regression for target X_i, and ω^{jj} is the jth diagonal element of the inverse correlation matrix of the predictors (Judge et al.,

1988, p. 243). (How to obtain the inverse correlation matrix of the predictors from the $\mathbf{R}^{-1}$ matrix in quadratic time is discussed in the next subsection.) The standard error $\hat{\sigma}_i$ can also be obtained in linear time from $\mathbf{R}^{-1}$ as follows.

Without loss of generality, we assume a zero mean and a unit standard deviation for all variables. Then $\sigma_i^2 = 1 - R_i^2$, where R_i^2 is the coefficient of determination of the regression for target X_i. This coefficient can be expressed as the scalar product of the $\mathbf{b}_i$ vector with the vector $\mathbf{r}_i$ of the pairwise correlation coefficients of the predictors with the target X_i (Raveh, 1985), which we read directly from the correlation matrix $\mathbf{R}$:

$$R_i^2 = \mathbf{b}_i^T \mathbf{r}_i.$$

An unbiased estimator $\hat{\sigma}_i$ for σ_i is then

$$\hat{\sigma}_i = \sqrt{\frac{n(1 - \mathbf{b}_i^T \mathbf{r}_i)}{n - d}}.$$

To sum up, we have a complexity of $\mathcal{O}(nd^3)$ to build and invert the correlation matrix, and $\mathcal{O}(d^3)$ to check for significance. This comes from having to obtain d times the inverse correlation matrix of $d-1$ predictors in $\mathcal{O}(d^2)$, and then checking their significance in linear time. The overall complexity of TC, including the collider identification and the constraint-propagation steps, is then $\mathcal{O}(nd^3 + d^2 2^{\alpha})$.

The weaknesses of this approach are its infeasability when the correlation matrix $\mathbf{R}$ does not have full rank (including the special case $n < d$, that is, when there are fewer samples than variables), the low power of the statistical tests with small data sets, and multicollinearity in the predictors. The symptoms of the last two points are that the t-tests do not refute the null hypothesis of zero weight because (i) there is not enough data to support it, or (ii) multicollinearity makes the weights lower than they should be, such that it becomes harder to interpret them as depicting the independent contribution of each predictor. We try to deal with this problem in the next section with the TC_{bw} algorithm.

4.2.1. SIGNIFICANCE TESTS

Independently of low sample sizes or multicollinearity, the statistical tests on the weights of the linear regression equations are a delicate point in TC. The choice of the Type I error rate α needs investigating as it significantly influences the result of the algorithm.

In a network of d nodes, the feature-selection step performs $d(d-1)/2$ tests to determine the undirected skeleton. We will falsely reject the null hypothesis $b_{ij} = 0$ about $m \cdot \alpha$ times on average, where $m < d(d-1)/2$ is the difference in the number of edges between the original DAG $\mathcal{G}_0$ and the complete graph. We will thus add on average $m \cdot \alpha$ wrong edges. We can set the significance level for the individual tests to be inversely proportional to $d(d-1)/2$ to avoid this problem (assuming a large m and thus rather sparse graphs), and check that it does not affect the Type II error rate too much, which we do now.

According to (10), the expression $(\hat{b}_{ij} - b_{ij})/\hat{\sigma}_{ij}$ follows a t distribution with $n - (d-1)$ degrees of freedom. If we call $\Psi(\cdot)$ the cumulative distribution function of a t distribution with $n - (d-1)$ degrees of freedom, we can write the Type II error rate β for each regression weight:

$$\beta_{ij} = \Psi(\Psi^{-1}(1 - \alpha/2) - |b_{ij}|/\hat{\sigma}_{ij}).$$

The values for $\hat{\sigma}_{ij}$ can be computed from the inverse correlation matrix $\mathbf{R}^{-1}$ and thus depend on the particular data set being analyzed, but the true b_{ij} are unknown. What we could do in theory to optimize α is to minimize the average number of extraneous (T_e) and missing (T_m) links:

$$T = T_e + T_m = m \cdot \alpha + \sum_{(i,j)\in\mathbf{E}} \beta_{ij},$$

where m is the number of edges missing in the original DAG compared to a full graph, and $\mathbf{E}$ is the set of arcs in the original DAG, so that $m + |\mathbf{E}| = d(d-1)/2$. As m, $\mathbf{E}$ and b_{ij} are unknown, we can only find an upper bound for the number of missed links T_m, provided (i) we can estimate the graph sparseness to approximate m; (ii) we assume $|b_{ij}| \geq \delta$; and (iii) we choose $\mathbf{E}^\star$ such that it maximizes the sum in (11), with $|\mathbf{E}^\star| = d(d-1)/2 - m$. Then we have:

$$T_m \leq \sum_{(i,j)\in\mathbf{E}^\star} \Psi(\Psi^{-1}(1-\alpha/2) - \delta/\hat{\sigma}_{ij}). \tag{11}$$

Although this bound was found too loose for practical use, we can model the Type I and Type II error rate as a function of α for artificial problems whose sparseness and regression weights are known. This is shown in Figure 3 for a specific instance of an Alarm data set (see Section 5 for details on this network) with two different sample sizes, $n = 50$ (left) and $n = 250$ (right). We did not use this information to tune α in the experiments, as it cannot be obtained without prior knowledge, but the curves showed that an α inversely proportional to $d(d-1)/2$ has the same order of magnitude as the optimal α on the data sets we analyzed.

What we also see is that the Type I error curve rapidly goes up, whereas the Type II error curve is upper-bounded by the total number of links in the original graph. In terms of pure number of errors, setting a low α will thus be more beneficial than setting a higher α to get a low β. It is worth discussing, however, depending on the particular problem to solve, which is more desirable: missing causal links or getting extra causal links. In terms of Bayesian networks, getting too few links prevents the model from being able to reconstruct the full joint probability distribution, because we lose the I-map property; whereas getting too many links implies having to estimate more parameters from the same data and thus complexifies a subsequent parameter learning task.

4.3. The TC_{bw} Algorithm

Despite correctness of TC, with a low number of samples n it fails to have enough evidence for rejecting the null hypothesis of zero regression weight, and thus misses links (see detailed results in Section 5), even for a high α. We now try to address this particular issue by successively eliminating the most insignificant predictors and reevaluating the remaining ones. This is actually a backward stepwise-regression method. Pseudocode for this heuristics is listed in Algorithm 7.

Intuitively, the problem to solve is that the regression weights cannot be high enough for significance with small sample sizes. By removing the most insignificant predictors and thus the most likely to be actually zero, we scale down the regression problem and increase the power of the tests. How many insignificant predictors to remove can be discussed; in our implementation, we compared $p = 1$ to $p = (\text{number of predictors})/2$ and found that the latter yielded results that were just as good with an important speed gain.

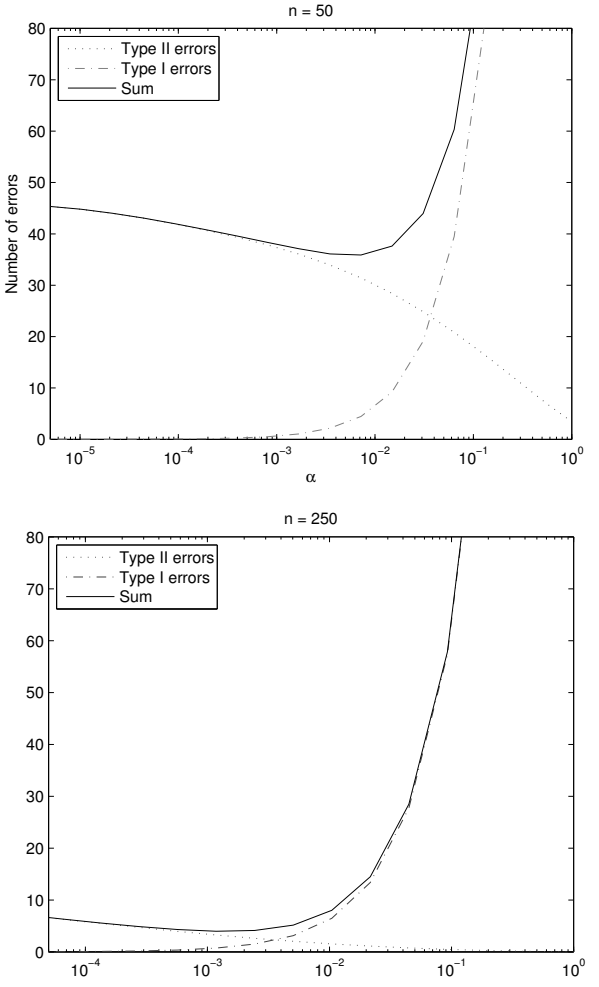


Figure 3: Expected Type I and II errors as a function of α

This stepwise regression raises some issues; notably, Tibshirani (1994) argues that the repeated tests on non-changing data are biased and that the remaining **b** coefficients are too large. We thus expect $\mathrm{TC}_{\mathrm{bw}}$ to be biased and to include more false positives than TC. Ideally, one would need a criterion to predict when the additional false positives would outweigh the benefits of reducing the false negatives. Whether such a criterion, which would allow us to know a priori whether TC or $\mathrm{TC}_{\mathrm{bw}}$ should be used, can be found, is an open question.

Solving a standard multiple regression problem with d predictors traditionally has complexity $\mathcal{O}(nd^3)$. Naïvely solving $d-1$ regression problems d times in the case $p=1$ would have a complexity of $\mathcal{O}(nd^5)$. But we can avoid reinverting matrices in the inner loop of the stepwise regression thanks to the following result.

Algorithm 7: The Total Conditioning Backward Feature-Selection Step

procedure TCBWFEATURESELECTION
X: the target variable to perform feature selection for
Input: D: $n \times d$ data set with n d-dimensional data points
Output: **S**: the set of selected variables

1 $\mathbf{P} \leftarrow \mathbf{V} \setminus X$ `/* all predictors */`
2 $\mathbf{S} \leftarrow \emptyset$ `/* significant predictors */`
3 **while** $\mathbf{P} \neq \emptyset$ **and** $\mathbf{P} \neq \mathbf{S}$ **do**
4 $\mathbf{b} \leftarrow$ weights of $\mathbf{P}$ in the problem of regressing X on $\mathbf{P}$
5 $\mathbf{S} \leftarrow \mathbf{S} \cup \{$predictors whose b weight is significant$\}$
6 $\mathbf{P} \leftarrow \mathbf{P} \setminus \{$the p less significant predictors$\}$
7 **end**
8 **return S**
end

Let $\Sigma = \mathbf{X}^T\mathbf{X}$ be n times the correlation matrix $\mathbf{R}$, where $\mathbf{X}$ is the $n \times d$ matrix representing a data set where all variables have zero mean and unit standard deviation. Then we can use Σ^{-1} to linearly find the weights of the regression problems and their standard error, which are needed for the t-tests. Suppose we find that variable X_1 is the weakest predictor, and want to reevaluate the weights of the other predictors at line 4 of TC_{bw}. Let $\mathbf{X}_{\setminus i}$ be the data set where variable X_i has been removed. Then we need the matrix Ω^{-1} to solve the new problem, where $\Omega = \mathbf{X}_{\setminus 1}^T\mathbf{X}_{\setminus 1}$. As a special case of Strassen's blockwise matrix inversion formula, we have:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \mathbf{c}^T \\ \mathbf{c} & \Omega \end{bmatrix}$$

$$\implies \Sigma^{-1} = \begin{bmatrix} \dfrac{1}{\sigma_{11} - \mathbf{c}^T\Omega^{-1}\mathbf{c}} & -\dfrac{\mathbf{c}^T\Omega^{-1}}{\sigma_{11} - \mathbf{c}^T\Omega^{-1}\mathbf{c}} \\ -\dfrac{\Omega^{-1}\mathbf{c}}{\sigma_{11} - \mathbf{c}^T\Omega^{-1}\mathbf{c}} & \Omega^{-1} + \dfrac{\Omega^{-1}\mathbf{c}\mathbf{c}^T\Omega^{-1}}{\sigma_{11} - \mathbf{c}^T\Omega^{-1}\mathbf{c}} \end{bmatrix}.$$

Let $\sigma^{ij} = (\Sigma^{-1})_{ij}$ and $\mathbf{b} = \Omega^{-1}\mathbf{c}$. Then $\mathbf{b}$ are the weights of the regression of X_1 on $X_2, \cdots, X_d$ and can be computed without knowing Ω^{-1} (Raveh, 1985), see (9). We have:

$$\sigma^{11} = 1/(\sigma_{11} - \mathbf{c}^T\mathbf{b})$$

and, $(\Sigma^{-1})_{\setminus 1}$ being the matrix Σ^{-1} where the first row and column have been removed,

$$(\Sigma^{-1})_{\setminus 1} = \Omega^{-1} + \mathbf{b}\mathbf{b}^T/(\sigma_{11} - \mathbf{c}^T\mathbf{b}).$$

We can thus compute Ω^{-1} given Σ^{-1} with complexity $\mathcal{O}(d^2)$ as follows:

$$\Omega^{-1} = (\Sigma^{-1})_{\setminus 1} - \sigma^{11}\mathbf{b}\mathbf{b}^T. \tag{12}$$

This trick is also used in TC to find the inverse correlation matrix of the predictors from the inverse correlation matrix of the whole variable set.

Equation (12) is implemented in TC_{bw} such that we never need to invert another matrix again once Σ^{-1} has been obtained, and leads to a complexity of $\mathcal{O}(d^2)$ for

stepwise elimination of a predictor. In the most computationally expensive case $p = 1$, this elimination of row and column of the inverse matrix is repeated at most $d - 2$ for each variable, yielding a complexity of $\mathcal{O}(nd^4)$ for the whole feature-selection step for all variables. The overall complexity of TC_{bw} is then $\mathcal{O}(nd^4 + d^2 2^\alpha)$. We are only adding one complexity degree in d with respect to TC with the additional stepwise regression.

5. Experimental Results

In this section, we report on experiments and results on two points separately. First, we test our procedure described in Algorithm 2 to recover the local structure with the collider set search given all Markov blankets, and compare it to the relevant steps of the GS algorithm, which are listed in Algorithm 1, with 5 different network topologies. For the sake of comparison, we also run the reference PC algorithm (Spirtes et al., 2001), initialized with the moral graph instead of the fully connected graph.

Second, we conduct experiments to investigate how the whole structure-learning algorithms behave. We first use the RFE-based approach. We then systematically compare TC, TC_{bw} and several reference algorithms, varying the data set size and the network size. Note that results for some algorithms may be sparser due to their prohibitive run times on some data sets.

5.1. Experimental Setup

In order to test the accuracy of the various algorithms, we chose to sample data from the following known networks, from the Bayes net repository (Elidan, 2001):

- Alarm network (Beinlich et al., 1989). This network has become a de facto standard benchmark for structure-learning algorithms: it contains 37 nodes, 46 arcs, 4 undirected in the PDAG of the equivalence class. It was originally designed to help interpret monitoring data to alert anesthesiologists to various situations in the operating room. It is depicted in Figure 4.

- Insurance (Binder et al., 1997), 27 nodes, 52 arcs, 18 undirected in its PDAG. It was designed to evaluate car insurance risks. This network has fewer nodes than Alarm but is denser, see Figure 5.

- Hailfinder (Abramson et al., 1996), 56 nodes, 66 arcs, 17 undirected in its PDAG. It is a normative system that forecasts severe summer hail in northeastern Colorado. See Figure 6.

- Carpo,[10] 61 nodes, 74 arcs, 24 undirected in its PDAG. It is meant to help diagnose the carpal tunnel syndrome. The version we used has three disconnected subgraphs, one of which is a single variable, and a relatively flat causal structure, as can be seen in Figure 7.

- A subset of Diabetes (Andreassen et al., 1991) with 104 nodes, 149 arcs, 8 undirected in its PDAG, which was designed as a preliminary model for insulin dose adjustment. This subset is made of 6 repeating patterns (there are 24 in the original network) of 17 nodes, plus 2 external nodes linked to every pattern. The first two of these patterns are shown in Figure 8.

10. Created by Alex Dagum with contributions from Mark Peot, as indicated on its page at the Bayes net repository. No corresponding publication was found.

We performed three series of experiments.

1. We compared our algorithm resolving the Markov blanket to the relevant steps of the Grow-Shrink algorithm, as described in Section 3.2, and to PC;
2. We tested the RFE-based approach and compared it to PC;
3. Finally, we compared TC and TC_{bw} to three reference algorithms and examine their accuracy, run time, and number of tests while varying the network structure, the network size, and the sample size.

The chosen reference algorithms are:

1. The PC algorithm. PC is, like TC and TC_{bw}, exponential in the worst case, when graphs are not sparse enough: we discuss which structural elements make PC or TC exhibit the exponential behavior;
2. The full Grow-Shrink algorithm, as described in Margaritis and Thrun (1999);
3. A state-of-the-art Bayesian structure-learning algorithm that works with continuous data sets, the Bach-Jordan scoring algorithm (Bach and Jordan, 2003), coupled with a greedy search in the space of DAGs. Note that Bayesian structure-learning algorithms are often score-based and return fully oriented DAGs. Maximizing the chosen score function might not minimize the number of structural errors as we report in these results.

For all simulation experiments, we generated the data sets by using the 5 graphs as a structure for a linear structural equations model: the parentless variables were sampled as Gaussians with zero mean and unit standard deviation; the other variables were defined as a linear combination of their parents with coefficients randomly distributed uniformly between 0.2 and 1, similarly to what was done in Scheines et al. (1995) for the evaluation of PC. The disturbance terms were also normally distributed. We compared the number of tests, the size of the conditioning sets, and the structural errors in case of runs with artificial data. A structural error is an arc addition, deletion, or reversal with respect to the original graph.

We used the implementation of PC proposed by Leray and François (2004) in the BNT Structure Learning Matlab package. The implementation of TC and TC_{bw} was also done in Matlab. The statistical tests were done using Fisher's z-transform of the partial correlation, unless otherwise stated. For PC and GS, we chose the default value of $\alpha = 0.05$; we note though that the optimal value of α is problem dependent and that especially with low sample sizes, hand tuning α can return better results than those listed here. For both TC and TC_{bw}, we set $\alpha = 2/(d(d-1))$, according to the discussion at the end of Section 4.2.

5.2. Local Structure Recovery with Markov Blanket Information

In this series of experiments, we compare RESOLVEMARKOVBLANKETS_COLLIDERSETS (CS) to RESOLVEMARKOVBLANKETS_GROWSHRINK and to a modified version of PC, where the graph being built is initialized with the moral graph (instead of the full graph in the original version of PC). This represents exactly the Markov blanket information available to the two other algorithms and allows a direct comparison. Note that we observe the PDAG that PC obtains *before* the constraint-propagation step building the

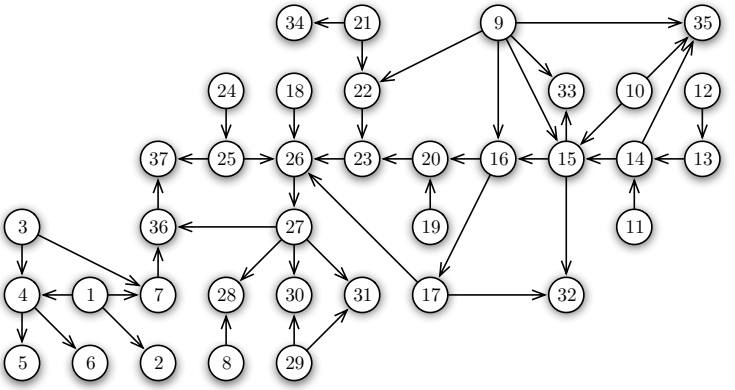


Figure 4: The Alarm network

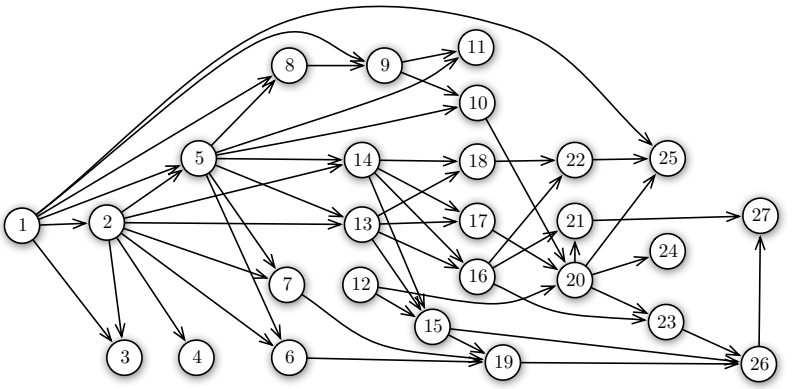


Figure 5: The Insurance network

maximally oriented PDAG, such that, in all three tested algorithms, we only expect the V-structures to be oriented.

We tested the three algorithms on each network using two methods to check for conditional independence: first, using a d-separation oracle with the original graph (which is equivalent to a perfect test); and second, using Fisher's z-transform of the sample partial correlation coefficient as computed on artificial data, with significance $\alpha = 0.05$. Using the oracle always yields correct graphs.

Table 1 shows the results of these experiments. We first list the results obtained when using a d-separation oracle to decide upon conditional independence. For GS, we ran two versions of Algorithm 1: one, which we name GS(1), where the subset searches at lines 4 and 12 proceed with decreasing sizes of the chosen subset $\mathbf{S}$, and another, GS(2), with increasing subset sizes. GS(1) usually leads to fewer tests, but with larger conditioning sets. The order of the subset searches for our method (lines 6 and 15 in Algorithm 2) was fixed to decreasing subset sizes, as this always led to fewer tests *and* smaller conditioning sets.

Table 1: Number of tests and size of the conditioning sets (noted $|\mathbf{Z}|$) as performed by various algorithms to recover the local network structure of the networks given perfect Markov blanket information. The star (*) notes PC results where the maximum size of the conditioning set has been set to 6 to avoid prohibitive run times.

Algorithm	Alarm	Insurance	Hailfinder	Carpo	Diabetes
modified PC					
# tests	11331	773572	19543985*	2025250*	93134*
avg. $\lvert\mathbf{Z}\rvert$	4.36	7.65	5.75*	5.47*	4.64*
max $\lvert\mathbf{Z}\rvert$	10	16	6*	6*	6*
GS(1)					
# tests	1485	6435	2809	209342	5414
avg. $\lvert\mathbf{Z}\rvert$	2.62	3.63	2.66	7.46	2.73
max $\lvert\mathbf{Z}\rvert$	8	11	7	15	10
GS(2)					
# tests	1472	7180	2979	200621	6197
avg. $\lvert\mathbf{Z}\rvert$	2.20	3.05	2.31	7.39	2.39
max $\lvert\mathbf{Z}\rvert$	7	8	7	15	8
CS					
# tests	214	1288	593	294	943
avg. $\lvert\mathbf{Z}\rvert$	1.80	2.69	2.30	1.79	2.13
max $\lvert\mathbf{Z}\rvert$	5	6	6	8	7

The results for the modified PC algorithm are only shown for the sake of comparison: PC is a general-purpose algorithm which is not specialized in such local structure recognition given the Markov blankets. What the comparison shows, however, is that, whenever this Markov blanket information is available or cheap to obtain, there are much more efficient approaches.

GS(1) and GS(2) are close to one another in all scores, and outperform PC (by several orders of magnitude) in the number of tests and (significantly) in average and

maximum size of the conditioning sets (except, artificially, for the results marked with a star), because it uses the Markov blanket information better. Our approach, however, is one order of magnitude better than GS(1) and GS(2) in terms of number of tests, while still using smaller average and maximum conditioning set sizes in all tested networks. Especially striking are the results on the Carpo network: this is an example where CS saves a lot of time ignoring the numerous links not part of triangles, whereas GS(1) and GS(2) also checks those, with the often large Markov blankets (Figure 7).

We then performed the same experiments, but using the statistical tests on data sampled from the networks as described in the previous sections. We used a fixed sample size $n = 500$ and averaged over 9 different samplings for each network. We only compared PC, GS(1) and CS on this series of experiments, preferring GS(1) to GS(2) because of the lower number of tests it usually performs. The exhaustive results are listed in Table 2 for the sake of completeness, and the sum of the structural errors is also shown in Figure 9 for easier visualization.

First, we see that we get similar results as in Table 1 as far as the number of tests and size of the conditioning sets are concerned: CS is faster and consistently performs fewer tests with smaller conditioning sets, which leads to an increased power of the tests. However, that is sometimes balanced out by the fact that CS relies on a single series of tests both to remove spouse links and to orient (possibly multiple) V-structures at the same time, thus leading to a greater penalty if the outcome of a test is wrong with respect to the initial graph.

We see that GS(1) and PC can beat CS on certain arc scores; PC, in particular, is good at avoiding arc orientation mistakes in these experiments. GS(1), which checks not only triangle links but all links to try to orient them, makes more orientation mistakes, especially on the Carpo network. PC tends to miss a few more arcs than CS, which in turn misses a few more than GS(1). But in total, CS beats GS(1) significantly on Insurance, Hailfinder, and Carpo, while performing slightly better on Alarm and being slightly outperformed on Diabetes. Based on these results, we will now use our collider set search as the method of choice to break up the Markov blankets for the next series of experiments.

5.3. RFE-Based Approach

In this series of experiments, we tested our RFE-based approach on the Alarm network with sample sizes $n = 100$, 200, 300, 400 and 500. Table 3 lists the results and shows the number of errors as measured at different stages of the algorithm:

1. Right after the Markov blanket identification, without adjustment. This compares the true Markov blanket of each variable with the identified Markov blanket as returned by Algorithm 5;

2. After building the moral graph. This notably excludes variables from Markov blankets if they do not satisfy the symmetry condition (2) due to the symmetry check performed at line 4 in the generic approach described in Algorithm 3;

3a. After removal of the spouse links using the Recursive Median (RM) algorithm (Margaritis, 2005) to check for conditional independence in the continuous domain;

3b. Alternatively, after removal of the spouse links using a test on Fisher's z-transform of partial correlation;

Table 2: Number of tests, size of the conditioning sets (noted $|\mathbf{Z}|$), and structural errors as returned by GS(1) and CS to recover the local network structure of the networks given perfect Markov blanket information. Results are given is the form "mean $\pm$ standard deviation over the 9 data sets." The best performer for each type of structural error has been highlighted in bold. All runs of PC were done with a forced maximum size of the conditioning set of 6. The dagger (†) notes PC results from a single data set instead of 9 because of the long completion times. Represented graphically in Figure 9.

Algorithm	Alarm	Insurance	Hailfinder	Carpo	Diabetes
mod. PC					
# tests	2850 ± 285	13461 ± 3247	$9681105^\dagger$	412791 ± 104080	57153 ± 9910
avg. $\|\mathbf{Z}\|$	2.97 ± 0.17	3.50 ± 0.33	$5.54^\dagger$	5.17 ± 0.15	4.37 ± 0.14
max $\|\mathbf{Z}\|$	6	6	$6^\dagger$	6	6
arcs:					
missing	5.44 ± 0.53	9.56 ± 1.01	$6^\dagger$	14.22 ± 1.64	9.56 ± 1.88
extra	0.33 ± 0.5	0.11 ± 0.33	$\mathbf{0}^\dagger$	$\mathbf{0.22} \pm 0.44$	$\mathbf{1.11} \pm 0.60$
reversed	**0**	$\mathbf{0.22} \pm 0.67$	$1^\dagger$	0.11 ± 0.22	$\mathbf{2.00} \pm 1.39$
TOTAL	5.78 ± 0.72	9.89 ± 1.47	$7^\dagger$	14.56 ± 1.86	12.67 ± 2.69
GS(1)					
# tests	1304 ± 60	4544 ± 195	2415 ± 63	129265 ± 17033	5239 ± 46
avg. $\|\mathbf{Z}\|$	2.66 ± 0.10	3.66 ± 0.04	2.62 ± 0.02	7.49 ± 0.08	2.76 ± 0.01
max $\|\mathbf{Z}\|$	8	11	7.89 ± 0.33	15	10
arcs:					
missing	$\mathbf{1.56} \pm 0.53$	$\mathbf{5.44} \pm 0.53$	$\mathbf{3.11} \pm 0.33$	**0**	6.11 ± 0.78
extra	0.56 ± 0.73	0.33 ± 0.71	1 ± 0.71	$\mathbf{0.22} \pm 0.44$	2.78 ± 1.64
reversed	1.11 ± 1.05	3.67 ± 2.12	8 ± 2.29	16.78 ± 2.49	2.67 ± 2.00
TOTAL	3.22 ± 1.81	9.44 ± 2.39	12.11 ± 2.74	17 ± 2.62	$\mathbf{11.55} \pm 3.03$
CS					
# tests	173 ± 3	782 ± 19	507 ± 18	308 ± 14.39	907 ± 4
avg. $\|\mathbf{Z}\|$	1.55 ± 0.03	2.36 ± 0.02	2.08 ± 0.03	1.90 ± 0.12	2.17 ± 0.01
max $\|\mathbf{Z}\|$	5	6	5	8	7
arcs:					
missing	$\mathbf{1.56} \pm 0.73$	6.33 ± 0.5	3.44 ± 0.52	**0**	$\mathbf{5.11} \pm 1.17$
extra	$\mathbf{0.44} \pm 0.53$	$\mathbf{0.22} \pm 0.44$	0.67 ± 0.70	0.33 ± 0.50	1.44 ± 1.51
reversed	0.11 ± 0.33	0.33 ± 0.5	$\mathbf{0.11} \pm 0.33$	**0**	7.11 ± 1.05
TOTAL	$\mathbf{2.11} \pm 0.96$	$\mathbf{6.89} \pm 1.03$	$\mathbf{4.22} \pm 1.27$	$\mathbf{0.33} \pm 0.50$	13.66 ± 2.20

4a. After removal of the spouse links using RM *and* after maximal orientation. This is actually the result that can be compared to other full structure-learning algorithms;

4b. After removal of the spouse links using partial correlation tests *and* after maximal orientation;

5. Finally, we show how PC performs on the same instance for comparison.

Note that the RM test is a Bayesian distribution-free conditional-independence test. In this case, where we use multivariate Gaussian distributed data, we do not expect it to perform better than the specialized z-test. We nevertheless include it in this series of experiments for two reasons. First, it allows the collider set search to be also distribution-free, in the sense that if "distribution-free feature selection" can be performed efficiently and consistently in the first phase, applying a subsequent collider set search does not make more assumptions on the distribution. Second, it allows to evaluate the cost of using a distribution-free algorithm on Gaussian data.

Detailed results are in Table 3 and the total number of structural errors is shown graphically in Figure 10. What we can read from the results is that, generally, the selected Markov blankets contain all variables from the true Markov blanket plus one or two additional variables. Starting at $n = 300$, on average, less than two variables were missed. Many spurious variable are selected, however, even for the larger data sets. This confirms the expectation the RFE approach also selects weakly relevant features: on average, the Markov blankets in the Alarm network have a size of 3.5, and on average 5.5 variables are selected per variable.

The symmetry check requiring Y to be part of $\mathbf{Mb}(X)$ and X to be part of $\mathbf{Mb}(Y)$ to add a link between X and Y fulfills its purpose, as even in the case $n = 200$ where on average about 73 variables enter wrong Markov blankets, only 4 extra links are added in the moral graph. As a side note, we thus argue that a global analysis can be beneficial to achieve better results on local tasks: we see here that determining via RFE the Markov blanket of a single variable is too inclusive, but that validating the selected variables globally, for instance with our Markov blanket symmetry check, allows to significantly reduce the number of false positives.

After the collider set search, the number of missing and extra arcs can both either increase or decrease. If the number of missing links increases, it is because the collider set search found d-separation too often while variables were actually dependent. If it decreases, it means that the missing arcs in the moral graph were spouse links, as their absence is not penalized in the PDAG any more. If the number of extra arcs increases, then the collider set search failed to identify spouse links; if it decreases, then the collider set search also removed through appropriate conditioning links that were not spouse links (which in turn possibly led to wrong orientations). Also, determining which part of the algorithm is responsible for a missed, extra, or reversed edge in a PDAG or CPDAG is not evident. As the feature-selection step is not alone responsible for the extra or missing links, the collider set search is not responsible for all orientation mistakes. In the collider set search, if a wrong spouse link is removed, it is because a wrong V-structure has been identified, so that the absence of an arc will be linked to the wrong orientation of the falsely recognized V-structure. It is also possible to construct cases where missing a variable in the feature-selection step will lead not only to a missing arc, but also to the detection of a spurious V-structure, even if all subsequent tests are perfect.

For the PDAGs obtained using z-tests, the number of missing arcs always decreases with respect to the moral graph, and so does the number of extra links for $n \geq 200$.

Table 3: Structural errors at various stages of the RFE-based approach, showing the missing, extra and reversed arcs with respect to the original graph. For Step 1, identification of the Markov blanket, the figures are averages over the 37 variables; that is, the count of the extra or missing variables per Markov blanket, and thus not directly comparable to the other steps. The sums of the errors for the CPDAGs are represented in Figure 10.

Stage	$n = 100$	$n = 200$	$n = 300$	$n = 400$	$n = 500$
1. Mb$(\cdot)$ ident.					
missing variables	17.33 ± 3.33	3.33 ± 1.48	0.78 ± 1.11	0.78 ± 1.48	0.33 ± 1.48
extra variables	80.66 ± 15.17	72.89 ± 9.25	74.78 ± 13.69	72.56 ± 9.99	73.33 ± 9.99
2. Moral graph					
missing arcs	23.44 ± 3.54	7.89 ± 3.48	4.33 ± 3.91	4.56 ± 3.47	4.33 ± 3.87
extra arcs	4.67 ± 2.24	4.11 ± 1.69	3.78 ± 1.20	3.11 ± 1.62	3.89 ± 2.20
TOTAL	28.11 ± 3.82	12.00 ± 2.55	8.11 ± 4.01	7.67 ± 4.06	8.22 ± 4.38
3a. PDAG/RM					
missing arcs	17.44 ± 2.35	10.00 ± 1.80	6.89 ± 2.67	6.33 ± 2.60	4.56 ± 1.51
extra arcs	4.78 ± 2.17	4.22 ± 1.56	3.33 ± 1.12	3.22 ± 1.48	3.44 ± 2.01
reversed arcs	1.22 ± 1.20	2.11 ± 1.27	3.11 ± 0.78	1.78 ± 0.97	0.67 ± 0.60
TOTAL	23.44 ± 1.67	16.33 ± 3.12	13.33 ± 2.65	11.33 ± 2.78	8.67 ± 2.37
3b. PDAG/z-t.					
missing arcs	11.89 ± 2.32	3.33 ± 1.32	2.67 ± 1.94	2.11 ± 1.17	2.56 ± 1.51
extra arcs	5.22 ± 2.22	4.00 ± 1.58	3.22 ± 1.20	2.78 ± 1.30	3.44 ± 2.01
reversed arcs	0.33 ± 0.50	0.56 ± 0.73	1.22 ± 0.67	0.78 ± 1.09	0.44 ± 1.13
TOTAL	17.44 ± 3.32	7.89 ± 2.15	7.11 ± 2.98	5.67 ± 2.12	6.44 ± 2.40
4a. CPDAG/RM					
missing arcs	17.44 ± 2.35	10.00 ± 1.80	6.89 ± 2.67	6.33 ± 2.60	4.56 ± 1.51
extra arcs	4.78 ± 2.17	4.22 ± 1.56	3.33 ± 1.12	3.22 ± 1.48	3.44 ± 2.01
reversed arcs	6.00 ± 3.87	8.67 ± 2.12	6.11 ± 2.32	6.11 ± 3.59	0.89 ± 0.60
TOTAL	28.22 ± 3.93	22.89 ± 3.02	16.33 ± 3.08	15.67 ± 2.24	8.89 ± 2.37
4b. CPDAG/z-t.					
missing arcs	11.89 ± 2.32	3.33 ± 1.32	2.67 ± 1.94	2.11 ± 1.17	2.56 ± 1.51
extra arcs	5.22 ± 2.22	4.00 ± 1.58	3.22 ± 1.20	2.78 ± 1.30	3.44 ± 2.01
reversed arcs	4.89 ± 3.33	4.33 ± 1.73	3.78 ± 1.09	3.00 ± 1.80	2.44 ± 1.13
TOTAL	22.00 ± 4.90	$\mathbf{11.67} \pm 2.40$	9.67 ± 2.87	$\mathbf{7.89} \pm 2.37$	8.44 ± 2.40
5. CPDAG/PC					
missing arcs	12.11 ± 2.52	7.44 ± 1.42	4.22 ± 0.97	5.67 ± 1.12	4.78 ± 0.83
extra arcs	4.56 ± 2.19	2.67 ± 1.87	2.78 ± 1.48	2.11 ± 0.93	2.00 ± 1.66
reversed arcs	2.67 ± 1.80	1.44 ± 1.51	1.22 ± 1.09	0.78 ± 1.09	0.67 ± 0.87
TOTAL	$\mathbf{19.33} \pm 4.66$	$\mathbf{11.56} \pm 3.2$	$\mathbf{8.22} \pm 2.11$	8.56 ± 1.59	$\mathbf{7.44} \pm 2.40$

We find that the more general RM test seems to return independence too often, as for $n \geq 200$ more links are missing in the PDAG than in the moral graph. (On highly nonlinear data, we would, however, expect RM to perform better than a z-test, which assumes Gaussianity.)

The CPDAGs do not have a number of adjacency errors different from their PDAGs; this step can only add directionality errors. We have nevertheless copied the results in order to improve the readability and to make the comparison with PC easier. Although the RFE approach can outperform PC in adjacency errors, PC still consistently makes fewer directionality errors. We remark, however, that the overall performance of RFE/SVR with z-tests is very comparable to that of PC, as also shown in Figure 10, which empirically justifies the intuition behind this approach.

5.4. TC and TC_{bw} vs. Competitors

For this series of experiments, we performed more systematic testing of TC, TC_{bw}, PC, the full GS and the Bach-Jordan method on data sets sampled from Alarm, Insurance, Hailfinder, Carpo, and Diabetes, varying the sample size. The Bach-Jordan method consists of a scoring function based on Mercer kernels coupled with a greedy search in the space of DAGs and was designed to learn Bayesian networks. It does not guarantee that the formal semantics of a causal graph are respected in the large-sample limit, but has been included in the experiments for the sake of comparison. Other possible competitors like SCA (Friedman et al., 2000) or AlgorithmMB (Peña et al., 2005) were inapplicable because generalizing them to handle continuous variables require techniques that are too computationally expensive, notably because of score-based subroutines that are hard to generalize.

The structural errors, like before, are missing, extra, and reversed arcs in the returned CPDAG with respect to the generating graph. For the Bach-Jordan method, similarly to what was done in Fu (2005), we converted the returned DAG to its essential graph first before checking for structural errors to avoid penalizing statistically equivalent structures. For all experiments, we also compare the run times and the number of tests performed by TC, TC_{bw}, GS, and PC.

Specific to the Bach-Jordan method is the issue of choosing the appropriate kernel parameters; that is, in our case, the σ width in the Gaussian kernel. Bach and Jordan (2003) claim that the algorithm is in general robust to the choice of σ. We have found, however, that for varying sample sizes, the number of structural errors is quite sensitive to σ. As the authors do not propose a heuristics to set it, we systematically tested the algorithm with $\sigma = 2$, 1, 0.5, and 0.3 for each run, and chose the outcome with the smallest sum of structural errors. In general, smaller data sets preferred $\sigma = 2$, while the larger ones preferred a smaller σ. The change of σ is not directly visible in the following plots of the errors, but it often leads to "zigzags" in the Bach-Jordan curves. This is due to the fact that we only tested a fixed number of values for σ and did not perform a full optimization of this parameter for each run. The results shown are thus not the best results obtainable with this method.

5.4.1. ALARM

The figures on p. 68 show the structural errors, run times and number of statistical tests against the number of samples for Alarm. For each sample size, 5 data sets were drawn from the model; the error bars picture the standard deviation over these 5 runs.

The numbers of extra and missing links seem to clearly decrease on average for all algorithms with an increasing number of samples, except for Bach-Jordan, which sometimes has the tendency to add more links when more data points are available. Note that Bach-Jordan's σ changes between the last two runs, explaining the abrupt change in the extra arcs. The number of reversed arcs seems to less satisfactory, in particular for TC. The explanation is that TC misses many arcs with low sample sizes, and thus does not actually get the opportunity to make many directionality errors for these cases. TC_{bw} exhibits a related behavior, although much less stronger. We also see that Bach-Jordan makes the most directionality errors (this is actually valid for all networks). GS reaches repeatedly a zero extra arc score for $n > 1000$, although it misses some more than the others.

Starting at about 200 samples, TC equals or outperforms PC, GS and Bach-Jordan. TC_{bw} beats both TC and PC, and the converging curves of TC and TC_{bw} show that the stepwise regression becomes unnecessary with about 400 samples. On average, TC was about 20 times faster than the implementation of PC we used, although the factor tended to decrease with larger sample sizes. TC_{bw} was naturally slower than TC, although only marginally compared to the speed difference with PC.

Overall, the constraint-based methods seem to perform approximately equally well for $n > 400$, and TC_{bw} and GS perform slightly better than PC for low sample sizes. Note that for low sample sizes, TC is always outperformed by TC_{bw}, PC, and GS, but is often the one to perform best when the sample size gets larger. The graphs in Figure 12 show that TC and TC_{bw} are fastest, although GS performed fewer tests that TC_{bw}.

5.4.2. INSURANCE

For this network, we find similar behaviors to Alarm, shown in the graphs on p. 69. The most striking difference is the clear tendency of Bach-Jordan to add more arcs when more data is available for this more densely connected network. Between the 5th and 6th sample sizes, there is again a change of σ. Comparing the curves of the missing and extra arcs, we see that this changes the tradeoff between false negatives and false positives.

In this case, too, TC_{bw} outperforms TC with low sample sizes (because it misses fewer arcs) but is outperformed with bigger data sets (because it adds too many). Both PC and GS, while being slightly better than TC_{bw} for $n < 100$, are outperformed starting at about $n = 500$. Note the overall good performance of GS in terms of arc orientation errors. The corresponding curve also decreases more smoothly with larger data sets. The Bach-Jordan method is unexpectedly fast on this data set, although poorly accurate. The pattern of the number of statistical tests is very similar to that of Alarm.

5.4.3. HAILFINDER

This network poses a problem to PC: we divided its run time and the number of tests by 10 in the graphs of Figure 16 on p. 70. Because of its long run times, PC was run only once for each point in the plots, so that the error bars are missing. PC runs into trouble because of the node cluster around variable 27 in the network (see Figure 6): it tries to separate it from the other nodes by doing subset searches on its large number of neighbors. In order to speed it up, we set the maximum node fan-in parameter to 6, so that PC would not attempt to conduct conditional-independence tests with conditioning sets larger than 6 (we see in Figure 16 how this imposes an upper bound on the run times of PC). TC and TC_{bw} do not run into this problem, because this cluster is correctly

left alone after the feature-selection step, done in $\mathcal{O}(d^3)$ operations. Note that TC, TC_{bw}, and GS *would* also spend a long time on this cluster if all neighbors of variable 27 were its parents, because they would contain a lot of extra spouse links to be checked with an exponential number of combinations. But this example shows that a local lack of sparseness is fatal to the efficiency of PC, whereas other algorithms can still deal with it if the density of the connections is caused by children rather than parents.

This network shows more clearly the missing arc problem that TC has with low sample size, and the benefits of using TC_{bw} rather than TC here, at least for $n < 2000$. On this network, GS performs overall well. It is beaten by TC only for $n > 2000$, but performs better than all others for $n < 200$. Bach-Jordan still exhibits the same tendency to add more arcs when more data is available. For this data set, σ changes twice, between the 4th and 5th, and between the 5th and 6th data set sizes. The 5th sample size seems to have generated an unfavorable data set for PC, as the number of extra arcs is particularly high.

Examining the run times designates TC as the fastest. This is important especially with larger sample sizes, as TC is often both the fastest and most accurate algorithm.

5.4.4. CARPO

The results for this network are shown on p. 71. The structural particularity of this network is multiple cases of a single variable having many children. PC performs overall badly on this network. For $n < 200$, GS is the clear winner: all other algorithms make many more errors. The plain TC especially misses many arcs. For $n > 500$, however, both TC and TC_{bw} slightly but consistently outperform GS. At $n = 800$, TC beats TC_{bw}. Bach-Jordan, although fast on this instance, adds again too many extra links, and makes numerous directionality errors.

5.4.5. DIABETES

This is our largest and final test network. The error patterns are most similar to those of the Insurance network, with the exception of Bach-Jordan, which performs more poorly here. We can detect two changes of σ: between the 3rd and 4th, and between the 5th and 6th sample sizes.

Starting at $n > 1000$, all constraint-based methods seem to yield similar overall accuracy. GS is better in terms of directionality errors; TC and TC_{bw} are better in terms of missed links. For $n > 4000$, TC and TC_{bw} have the same accuracy and slightly beat GS and PC, while they are beaten significantly for $n < 800$. We note that the extra links added by GS seem to allow it to obtain a better directionality accuracy than in our first series of experiments, where it was given the exact moral graph as input.

5.4.6. DISCUSSION

Both TC and TC_{bw} slightly but consistently beat the other competitors when the sample size exceeds one or two thousand, depending on the network. They are usually weaker with low sample sizes because of missed arcs. GS beats TC with small data sets, because of the way that PC goes through conditioning sets for the statistical tests (Tsamardinos et al., 2006, discuss in detail this particular issue in the case of tests with discrete variables). The score-based Bach-Jordan method was found difficult to tune with the parameter σ. For this multivariate Gaussian case, its performance is usually worse than the other tested algorithms. This also reflects the fact PC, GS, TC and TC_{bw} with

z-tests are all "tuned" for multivariate Gaussian data. The additional errors made by Bach-Jordan reflect the price of being more generic.

In terms of run time, PC is slowed down by nodes with a high degree, whereas TC or GS handle them without the exponential time complexity growth if they are not part of triangles, as in Hailfinder. In general, TC and TC_{bw} resolve all conditional-independence relations (up to married parents) in the feature-selection step in $\mathcal{O}(d^3)$ and $\mathcal{O}(d^4)$, respectively, whereas all PC can do in $\mathcal{O}(d^{2+\alpha})$ is resolve conditional-independence relations with conditioning sets of cardinality α. It is then reasonable to expect algorithms like GS, TC and TC_{bw} to scale better than PC on sparse networks where nodes have a small number of parents. The exponential growth in PC can be seen in case the nodes have a high degree, be it parents or children; in TC and GS, it is due to large fully-connected triangle structures and to spouse links coming from the Markov blanket-construction step. And whereas these large structures imply a high degree, the converse is not true (for instance in the Hailfinder network). So, PC will exhibit an exponential behavior on all problem instances where TC and GS also exhibits this behavior, but the converse is not true.

It is interesting to investigate what kind of high-degree structure is more likely to appear. If it is a node with many children (as node 27 in Hailfinder), which we call an *explosion pattern*, TC can handle it efficiently. If it is a node with many parents, an *implosion pattern*, then none of these algorithms can recognize it in polynomial time. Explosion patterns correspond to a single cause that has many effects; implosion patterns correspond to many causes leading to the same effect. It remains open for discussion to know which one is more likely to occur with real-life data sets.

GS could not be beaten on small sample sizes. It is yet an unsolved challenge for TC and TC_{bw} to handle problems where the number of variables exceeds the number of samples, as in gene expression networks, thus leading to an attempt at inverting a matrix that does not have full rank. Regularizing the covariance matrix might help make TC_{bw} more robust in this case. Computationally, TC_{bw} does add a degree of complexity with respect to TC, and the number of tests that TC_{bw} performs is usually comparable to GS.

TC_{bw} helps solving problems with TC and small data sets, but still cannot operate below the $n = d$ threshold. The exact sample size where TC_{bw} stops performing better than TC does not appear to be a simple function of the n or d but depends on the structure of the network. It would be useful to investigate when the feature-selection addition of TC_{bw} becomes irrelevant. And as GS is more accurate with small sample sizes, finding a similarly testable condition predicting the threshold where TC is more accurate than GS would allow to merge the approaches into a single algorithm that knows which Markov blanket approach to use in order to achieve better results.

6. Conclusion

Causal discovery and feature selection are closely related: optimal feature selection discovers Markov blanket as sets of strongly relevant features, and causal discovery discovers Markov blankets as direct causes, direct effects, and common causes of direct effects. By performing perfect feature selection on each variable, we get the undirected moral graph as an approximation of the causal graph. An extra step, the collider set identification, is needed in order to transform the Markov blankets into parents, children, and spouses. This step is exponential in the worst case, but is actually efficient provided the graph is sparse enough—a common assumption of many algorithms. We

proposed an algorithm to do this task and compared it favorably to the similar steps of the Grow-Shrink algorithm.

Determining the Markov blanket with existing backward feature elimination like RFE eliminates the irrelevant variables in the large sample limit, but remains too inclusive. Global corrections have to be made, such as for instance insuring that a variable in the selected Markov blanket of a target also includes this target in its own selected Markov blanket. We conducted experiments that confirmed that this adjustment discards most false positives, and thus provided a hint that the approach is consistent in the large-sample limit. The main challenge is to perform feature selection for all variables in an efficient way. This task is tractable with the multivariate Gaussian assumption. We presented the TC and the TC_{bw} algorithms, which fit into the described framework, and compared them to PC, GS, and a Bayesian structure-learning method. For small sample sizes, GS usually makes fewer structural errors, and TC/TC_{bw} are better for larger samples sizes.

We are convinced of the superiority of the Markov blanket approaches as described in this paper. We invoke as support for this claim the high run times of PC, and the good low and high sample size accuracy of GS and TC/TC_{bw}, respectively. Not only are Markov blanket techniques much more scalable, they can be more accurate; they are also more easily modifiable to construct only parts the network deemed relevant by some criterion.

The biggest challenges we face now with causal structure learning include robust and consistent distribution-free structure learning with continuous and potentially highly nonlinear data. In the future, we intend to make use of this framework to develop such techniques and thus try to get rid of the Gaussianity assumption, often impractical with real-life data sets.

Acknowledgments

We would like to thank Dimitris Margaritis, who kindly provided us with his C implementation of the Recursive Median conditional-independence test; and Francis Bach and Lawrence Fu, who provided us with a Matlab/C implementation of the Bach-Jordan algorithm. We also thank the anonymous reviewers for their helpful comments and pointers, which led to a significantly enhanced version of this paper.

Appendix A.

For all proofs, we assume the given data set D is faithful.

Lemma 12 *In a DAG $\mathcal{G}$, any (undirected) path π of length $\ell(\pi) > 2$ can be blocked by conditioning on any two consecutive nodes in π.*

Proof It follows from Definition 5 that a path π is blocked when either at least one collider (or one of its descendants) is not in the conditioning set **S**, or when at least one non-collider is in **S**. It therefore suffices to show that conditioning on two consecutive nodes always includes a non-collider. This is the case because two consecutive colliders would require bidirected arrows, which is a structural impossibility with simple DAGs. ■

Lemma 13 *In a DAG $\mathcal{G}$, two nodes X, Y are* d*-connected given all other nodes $\mathbf{S} = \mathbf{V} \setminus \{X, Y\}$ if and only if any of the following conditions holds:*

(i) There is an arc from X to Y or from Y to X (i.e., $X \to Y$ or $X \leftarrow Y$);
(ii) X and Y have a common child Z (i.e., $X \to Z \leftarrow Y$).

Proof We prove this by first proving an implication and then its converse.

($\Longleftarrow$) If (*i*) holds, then X and Y cannot be *d*-separated by any set. If (*ii*) holds, then Z is included in the conditioning set and *d*-connects X and Y by Definition 5.

($\Longrightarrow$) X and Y are *d*-connected given a certain conditioning set when at least one path remains open. Using the conditioning set $\mathbf{S}$, paths of length > 2 are blocked by Lemma 12 since $\mathbf{S}$ contains all nodes on those paths. Paths of length 2 contain a mediating variable Z between X and Y; by Definition 5, $\mathbf{S}$ blocks them unless Z is a common child of X and Y. Paths of length 1 cannot be blocked by any conditioning set. So the two possible cases where X and Y will be *d*-connected are (*i*) or (*ii*). ■

Corollary 14 *Two variables X, Y are dependent given all other variables $\mathbf{S} = \mathbf{V} \setminus \{X, Y\}$ if and only if any of the following conditions holds:*

(i) X causes Y or Y causes X;
(ii) X and Y have a common effect Z.

Proof It follows directly from Lemma 13 due to the faithful structure, which ensures that there exists a DAG where conditional independence and *d*-separation map one-to-one. Lemma 13 can then be reread in terms of conditional independence and causation instead of *d*-separation and arcs. ■

Property 7 (Total conditioning) *In the context of a faithful causal graph $\mathcal{G}$, we have:*

$$\forall X, Y \in \mathbf{V} : \big(X \in \mathbf{Mb}(Y) \iff (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\})\big).$$

Proof This is a direct consequence of Corollary 14, where points *(i)* and *(ii)* lead to the definition of the Markov blanket of Y as *(i)* all its causes and effects, and *(ii)* the other direct causes of its effects. This is equivalent to $\mathbf{Mb}(Y)$ in $\mathcal{G}$. ■

Lemma 15 *When it exists, the subset $\mathbf{Z}$ that has the Collider Set property for the pair (X, Y) is the set of all direct common effects of X and Y.*

Proof We prove this using $\mathbf{Z}$ and a corresponding $\mathbf{S}_{XY}$ that fulfills (7).

($\Longrightarrow$) ($Z_i \in \mathbf{Z} \implies X \twoheadrightarrow Z_i \twoheadleftarrow Y$.) By (7) and (8), we know that each Z_i opens a dependence path between X and Y (which are independent given $\mathbf{S}_{XY}$) by conditioning on $\mathbf{S}_{XY} \cup \{Z_i\}$. By Definition 5, conditioning on Z_i opens a path if Z_i is either a colliding node or one of its descendants. As, by definition, $\mathbf{Z} \subseteq \mathbf{Tri}(X - Y)$, we are in the first case. We conclude that Z_i is a direct effect of both X and Y.

($\Longleftarrow$) ($X \twoheadrightarrow Z_i \twoheadleftarrow Y \implies Z_i \in \mathbf{Z}$.) Note that (7) and (8) together are implied in presence of a V-structure $X \twoheadrightarrow Z_i \twoheadleftarrow Y$. Thus, a direct effect is compatible with the conditions. The fact that $\mathbf{Z}$ captures all direct effects follows from the maximization of

its cardinality. ■

Lemma 9 *In the context of a faithful causal graph, the set* $\mathbf{Z}$ *that has the Collider Set property for a given pair* (X, Y) *exists if and only if* X *is neither a direct cause nor a direct effect of* Y*, and is unique when it exists.*

Proof The fact that $\mathbf{Z}$ exists if and only if X is neither a direct cause nor a direct effect of Y is a direct consequence of (7), which states that X and Y can be made conditionally independent. This is in contradiction with direct causation.

We now show unicity, using interchangeably the criteria of d-separation and conditional independence, as allowed by the Faithfulness assumption. Suppose that, for a pair (X, Y), two sets $\mathbf{Z}$, $\mathbf{W}$ have been found that fulfill the Collider Set property, with the corresponding d-separating sets $\mathbf{S}_{XY}^{\mathbf{Z}} \subseteq \mathbf{V} \setminus \{X, Y\} \setminus \mathbf{Z}$ and $\mathbf{S}_{XY}^{\mathbf{W}} \subseteq \mathbf{V} \setminus \{X, Y\} \setminus \mathbf{W}$ fulfilling (7). Let $\mathbf{Z}^\star = \mathbf{Z} \setminus \mathbf{W}$. Due to symmetry, proving that $\mathbf{Z}^\star$ is empty proves that $\mathbf{Z} = \mathbf{W}$.

Suppose that $\mathbf{Z}^\star \neq \emptyset$; that is, $\exists Z \in \mathbf{Z}^\star$. Then, by definition, we have that $\left(X \perp Y \mid \mathbf{S}_{XY}^{\mathbf{Z}}\right)$ and $\left(X \not\perp Y \mid \mathbf{S}_{XY}^{\mathbf{Z}} \cup \{Z\}\right)$. We now have two cases: either (i) $Z \notin \mathbf{S}_{XY}^{\mathbf{W}}$, or (ii) $Z \in \mathbf{S}_{XY}^{\mathbf{W}}$. In the former case (i), consider the set $\mathbf{W}' = \mathbf{W} \cup \{Z\}$. Then $\mathbf{W}'$ also fulfills the Collider Set property with the same d-separating set $\mathbf{S}_{XY}^{\mathbf{W}}$: the only additional condition is $\left(X \not\perp Y \mid \mathbf{S}_{XY}^{\mathbf{W}} \cup \{Z\}\right)$. This holds because, as shown by Lemma 15, Z is a direct child of X and Y, and conditioning on it opens a path, no matter what the conditioning set is. But all this is in contradiction with the definition stating that any set fulfilling this property must be the largest set to do so, because the cardinality of $\mathbf{W}'$ is greater than that of $\mathbf{W}$.

In the latter case (ii), the d-separating set $\mathbf{S}_{XY}^{\mathbf{W}}$ contains Z. But this is impossible due to the same reason that Z is a direct child of both X and Y and that thus any set containing Z cannot d-separate X and Y. We therefore conclude $\mathbf{Z}^\star = \emptyset$ and $\mathbf{Z} = \mathbf{W}$, which leads to the uniqueness of the set fulfilling the Collider Set property. ■

Theorem 10 *In the large sample limit, for faithful, causally sufficient data sets, the procedure* RESOLVEMARKOVBLANKETS_COLLIDERSETS *correctly identifies all V-structures and all spouse links, assuming consistent statistical tests.*

Proof First, we note that in a moral graph, a node X is connected to its parents, children, and spouses. Thus, all spouse links to be removed are in the moral graph, and, by the definition of spouse, each spouse link between X and Y corresponds to at least one unshielded collider for the pair (X, Y). Additionally, by the definition of unshielded collider, X and Y are nonadjacent, so that for each spouse link $X - Y$ there is a set $\mathbf{S}_{XY}$ such that $\left(X \perp Y \mid \mathbf{S}_{XY}\right)$ by the contraposition of (4). So, when such a set $\mathbf{S}_{XY}$ is found, the link $X - Y$ is removed, and for each Z such that $X - Z - Y$ and $Z \notin \mathbf{S}_{XY}$, we orient the triplet as $X \to Z \leftarrow Y$ for the exact same reason that allows IC (or PC) to do the same in Step 2 of the algorithm (Pearl, 2000). The proof boils down to proving that the proposed search procedure always identify a d-separating set $\mathbf{S}_{XY}$ when there is one.

If some S_{XY} exists, then the link between X and Y is a spouse link by definition of a moral graph, which implies that X and Y have a nonempty set of common effects $\mathbf{Z}$. Each $Z \in \mathbf{Z}$ is linked to both X in Y and is thus in $\mathbf{Tri}(X - Y)$ by definition. Let us assume we can d-separate X and Y by some set: then, by the definition of d-separation, only conditioning on a common effect or a descendant of a common effect can create a

dependency. In Algorithm 2, all possible colliders (line 6) and descendants of currently conjectured colliders (line 13) undergo a subset search, such that there will always be one iteration where all colliders and their descendants will be left out of the conditioning set. It is then enough to show that all d-connecting paths between X and Y that are not due to conditioning on a collider or collider's descendant go through the base conditioning set as determined at line 5.

To prove this, we note that the subset search at line 6 will always go through an iteration where it blocks all such d-connecting paths of length 2, that is, patterns of the type $X \to W \to Y$ and $X \leftarrow W \to Y$. As a direct consequence of the fact that we are working on the moral graph, all longer dependency paths go both through a node W in the set of immediate neighbors $\mathbf{Bd}(X)$ of X, and through a node in $\mathbf{Bd}(Y)$. Let us look at $\mathbf{Bd}(X)$. We have two cases: either (i) $W \in \mathbf{Tri}(X - Y)$ and will eventually be blocked by the subset search at line 6, or (ii) $W \in \mathbf{Bd}(X) \setminus \mathbf{Tri}(X - Y)$ (and thus $W \in \mathbf{Bd}(X) \setminus \mathbf{Tri}(X - Y) \setminus \{Y\}$ because $W \neq Y$). This set is exactly the set selected as base conditioning set at line 5, blocking all such paths, up to some symmetry with Y. The fact that we may choose the smaller of the two possible base conditioning sets is due to symmetry reasons. ■

Theorem 16 *If the variables are jointly distributed according to a multivariate Gaussian, TC returns the maximally oriented PDAG of the Markov equivalence class of the DAG representing the causal structure of the data-generating process in the large sample limit, assuming statistically consistent tests.*

Proof An edge is added between X and Y in the feature selection if we find that $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}} \neq 0$. We conclude $(X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\})$ owing to the multivariate Gaussian distribution. Corollary 14 says that this implies that X causes Y or Y causes X, or that they share a common child. Therefore, each V-structure is turned into a triangle by the end of the feature-selection step. The collider set search then examines each link $X - Y$ part of a triangle, and by Lemma 15, we know that if the search for a set $\mathbf{Z}$ that has the Collider Set property succeeds, there must be no link between X and Y. We know by the same lemma that this set includes all colliders for the pair (X, Y), so that all V-structures are correctly identified. Step 3 is the same as in the IC or PC algorithms; see Pearl and Verma (1991) and Spirtes et al. (2001). ■

References

B. Abramson, J. Brown, A. Murphy, and R. L. Winkler. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12:57–71, 1996.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON, a novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, pages 21–25, 2003.

S. Andreassen, R. Hovorka, J. Benn, Kristian G. Olesen, and E. R. Carson. A model-based approach to insulin adjustment. In *Proc. of the Third Conf. on AI in Medicine*, pages 239–248. Springer-Verlag, 1991.

K. Baba, R. Shibata, and M. Sibuya. Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, 46(4), 2004.

F. R. Bach and M. I. Jordan. Learning graphical models with Mercer kernels. In *Advances in Neural Information Processing Systems 15*, 2003.

I. 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 *Proc. of the Second European Conf. on AI in Medicine*, pages 247–256, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29, 1997.

D. M. Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2002.

G. Elidan. Bayes net repository. Website, 2001. URL http://compbio.cs.huji.ac.il/Repository/.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. In *RECOMB*, pages 127–135, 2000.

L. D. Fu. A comparison of state-of-the-art algorithms for learning Bayesian network structures from continuous data. Master's thesis, Vanderbilt University, 2005.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

I. Guyon, C. Aliferis, and A. Elisseeff. Causal feature selection. In H. Liu and H. Motoda, editors, *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press, 2007.

D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear SVM-based feature selection. In *Proceedings of the Twenty First International Conference on Machine Learning*, 2004.

D. M. Hausman and J. Woodward. Independence, invariance and the causal Markov condition. *British Journal for the Philosophy of Science*, 50:521–583, 1999.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, 1994.

G. G. Judge, R. Carter Hill, W. E. Griffiths, H. Lütkepohl, and T.-C. Lee. *Introduction to the Theory and Practice of Econometrics, 2nd Edition*. Wiley, 1988.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

P. Leray and O. François. BNT structure learning package, 2004. URL http://banquiseasi.insa-rouen.fr/projects/bnt-slp/.

D. Margaritis. Distribution-free learning of Bayesian network structure in continuous domains. In *Proc. of the 20th National Conf. on AI*, 2005.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12*, 1999.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér. Consistent feature selection for pattern recognition in polynomial time. *The Journal of Machine Learning Research*, 8: 589–612, 2007.

J. M. Peña, J. Björkegren, and J. Tegnér. Scalable, efficient and correct learning of Markov boundaries under the faithfulness assumption. In *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 136–147, 2005.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Los Altos, 1988.

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995.

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

J. Pearl and T. Verma. A theory of inferred causation. In *Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1991.

J.-P. Pellet and A. Elisseeff. A partial correlation-based algorithm for causal structure discovery with continuous variables. In *7th International Symposium on Intelligent Data Analysis*, 2007.

A. Raveh. On the use of the inverse of the correlation matrix in multivariate data analysis. *The American Statistician*, 39:39–42, 1985.

R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The TETRAD project: Constraint based aids to causal model specification. Technical report, Carnegie Mellon University, Dpt. of Philosophy, 1995.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical report, NeuroCOLT2, 1998.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, Second Edition*. The MIT Press, 2001. ISBN 0262194406.

A. Statnikov, D. Hardin, and C. F. Aliferis. Using SVM weight-based methods to identify causally relevant and non-causally relevant variables. Technical report, Vanderbilt University, USA, 2006.

D. Steel. Homogeneity, selection, and the faithfulness condition. Technical report, Michigan State University, Department of Philosophy, 2005.

M. Talih. *Markov Random Fields on Time-Varying Graphs, with an Application to Portfolio Selection*. PhD thesis, Hunter College, 2003.

R. Tibshirani. Regression shrinkage and selection via the lasso. Technical report, University of Toronto, 1994.

I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy. *Artificial Intelligence and Statistics*, 2003.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In ACM Press, editor, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678, 2003.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 2006.

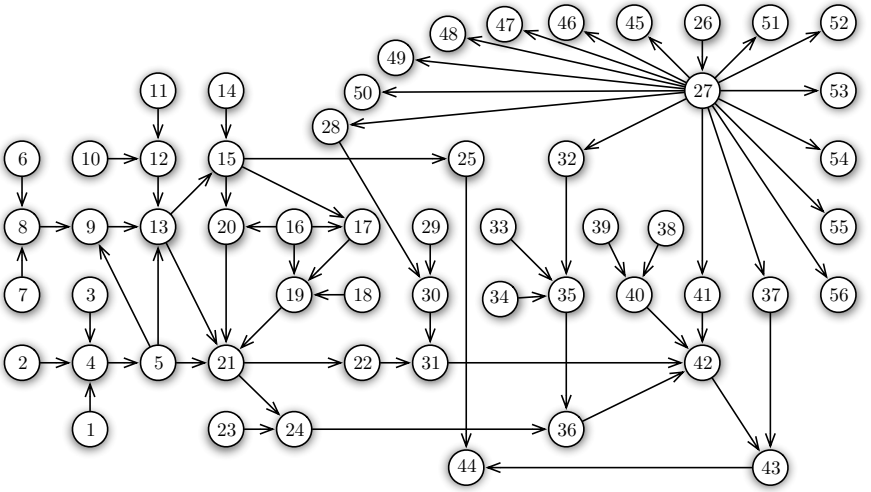


Figure 6: The Hailfinder network

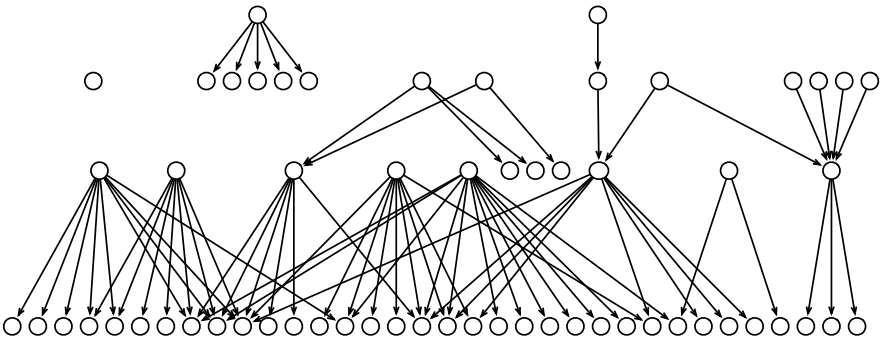


Figure 7: The Carpo network

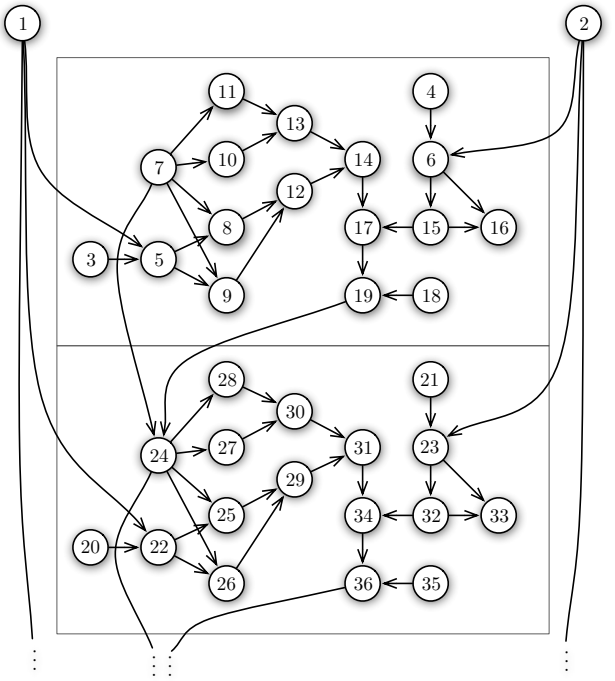


Figure 8: Two of the six patterns of the Diabetes network

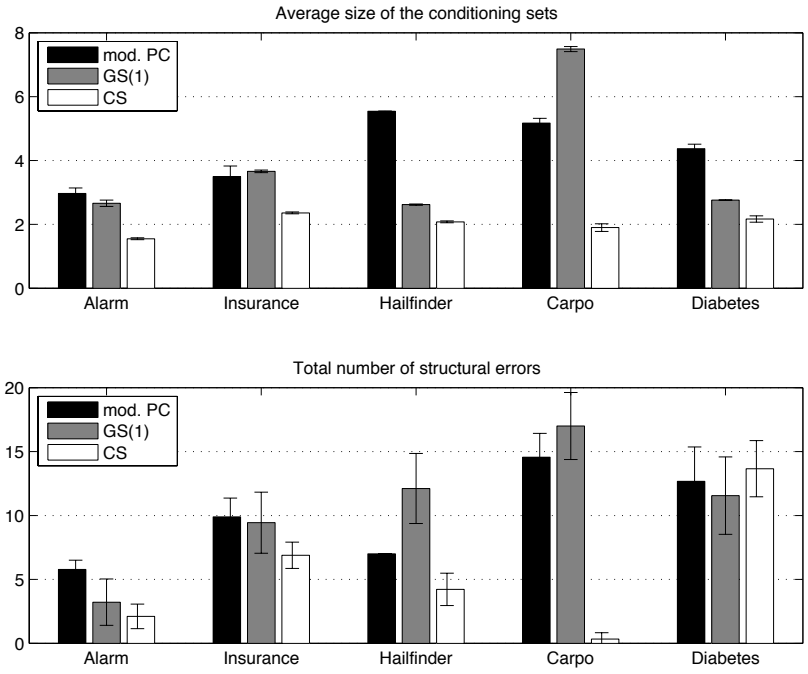


Figure 9: Average size of the conditioning sets and total number of errors for the three local structure discovery algorithms on various networks. Graphical representation corresponding to the results in Table 2.

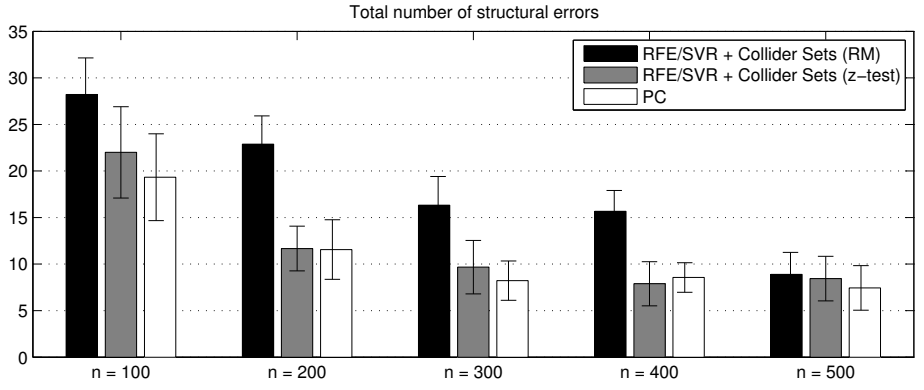


Figure 10: Total number of errors for the CPDAGs returned by the three global structure discovery algorithms on the Alarm network with various sample sizes. Graphical representation corresponding to the results in Table 3.

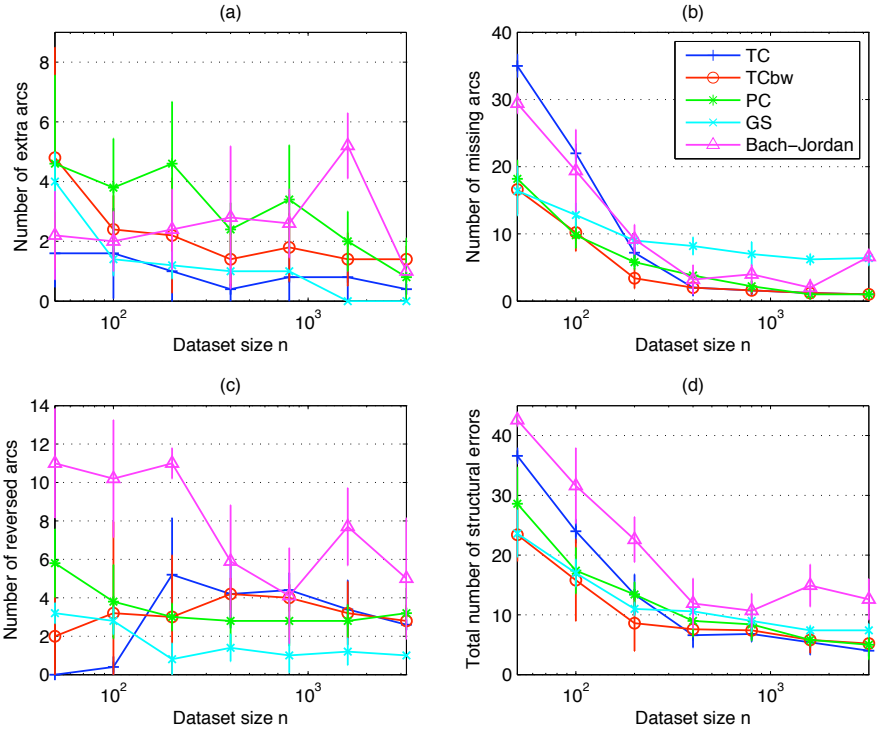


Figure 11: Differentiated errors on Alarm as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

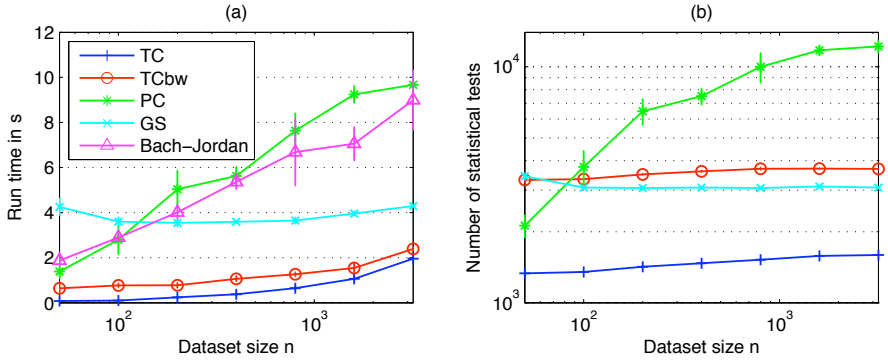


Figure 12: Alarm: (a) run times and (b) number of statistical tests as a function of the sample size n.

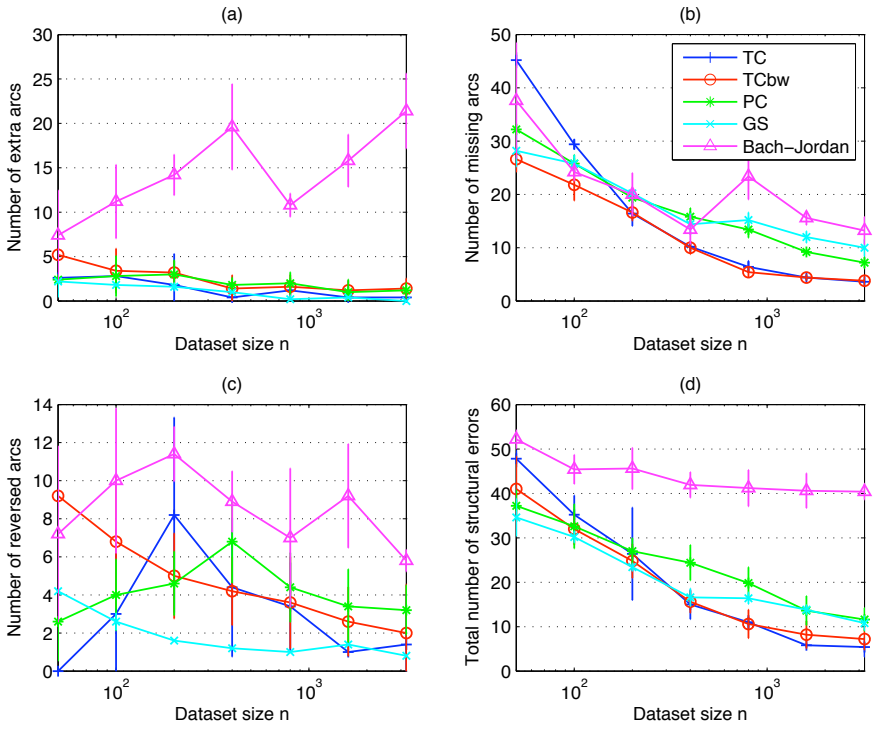


Figure 13: Differentiated errors on Insurance as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

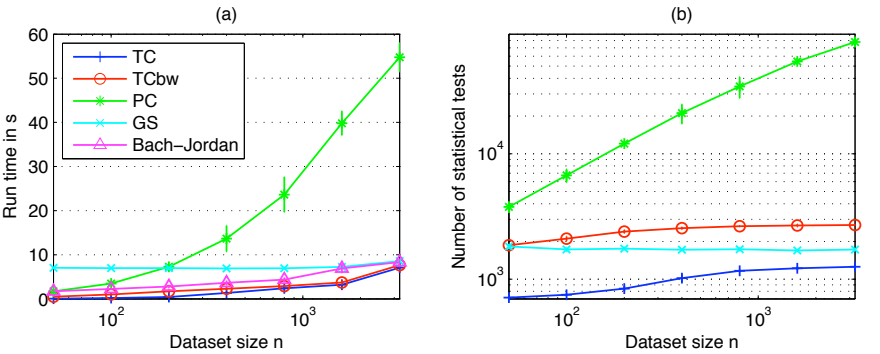


Figure 14: Insurance: (a) run times and (b) number of statistical tests as a function of the sample size n.

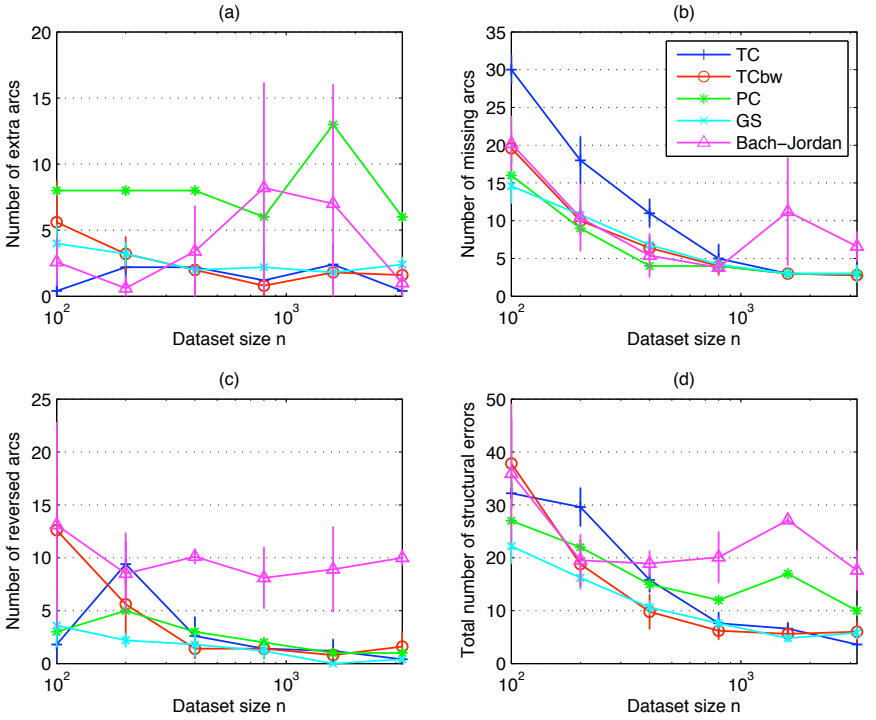


Figure 15: Differentiated errors on Hailfinder as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

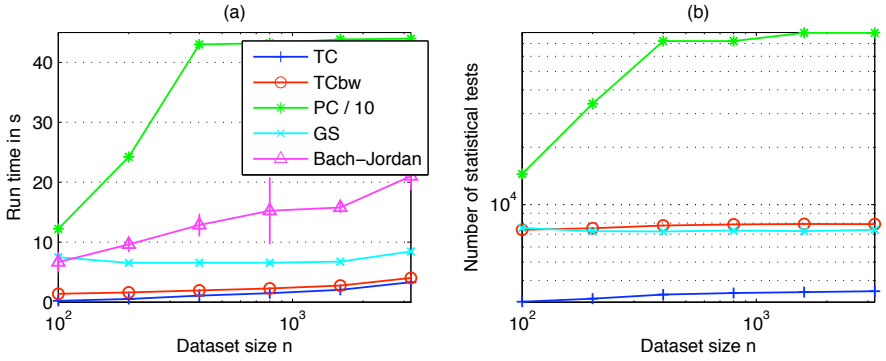


Figure 16: Hailfinder: (a) run times and (b) number of statistical tests as a function of the sample size n.

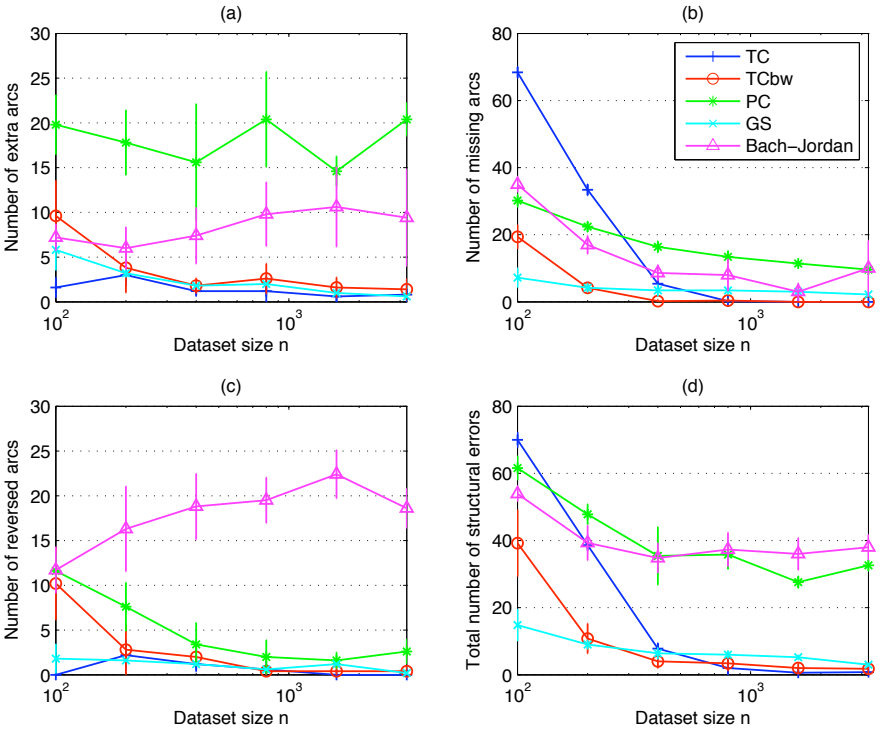


Figure 17: Differentiated errors on Carpo as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

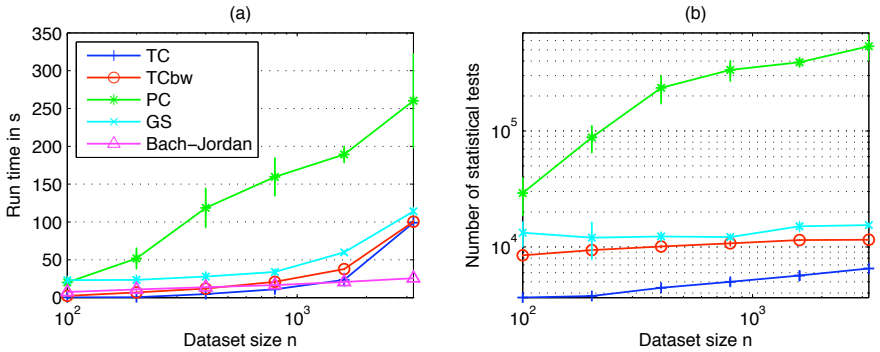


Figure 18: Carpo: (a) run times and (b) number of statistical tests as a function of the sample size n.

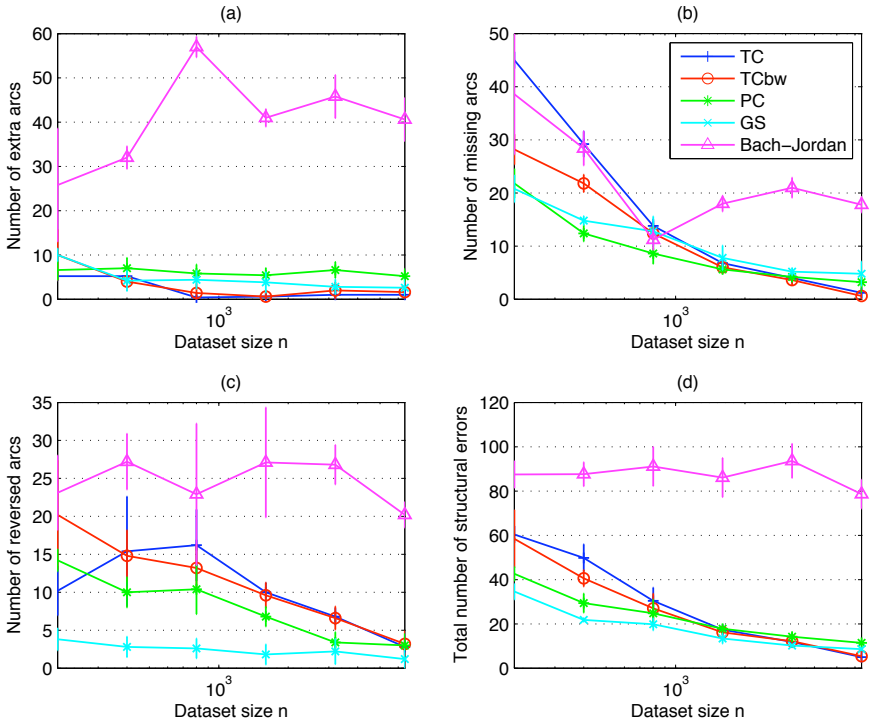


Figure 19: Differentiated errors on Diabetes as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

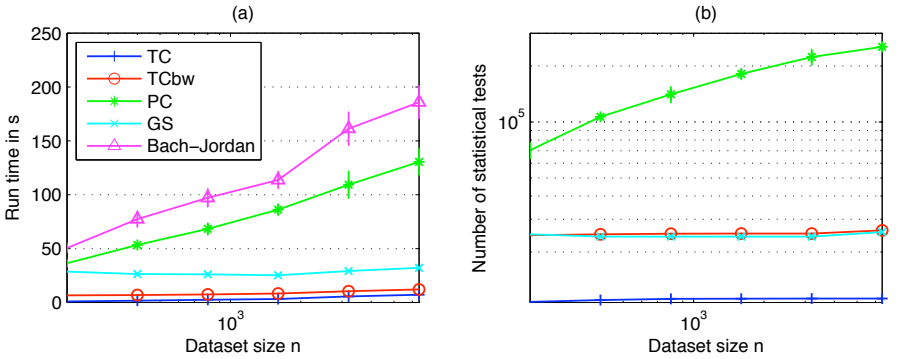


Figure 20: Diabetes: (a) run times and (b) number of statistical tests as a function of the sample size n.

JMLR: Workshop and Conference Proceedings 9:1437–1474, 2008 Submitted 6/07; Revised 2/08; Published 7/08

Causal Reasoning with Ancestral Graphs

Jiji Zhang JIJI@HSS.CALTECH.EDU
Division of the Humanities and Social Sciences
California Institute of Technology
Pasadena, CA 91106, USA

Editor: Gregory F. Cooper

Abstract

Causal reasoning is primarily concerned with what would happen to a system under external interventions. In particular, we are often interested in predicting the probability distribution of some random variables that would result if some other variables were *forced* to take certain values. One prominent approach to tackling this problem is based on causal Bayesian networks, using directed acyclic graphs as *causal* diagrams to relate post-intervention probabilities to pre-intervention probabilities that are estimable from observational data. However, such causal diagrams are seldom fully testable given observational data. In consequence, many causal discovery algorithms based on data-mining can only output an equivalence class of causal diagrams (rather than a single one). This paper is concerned with causal reasoning given an equivalence class of causal diagrams, represented by a (partial) *ancestral graph*. We present two main results. The first result extends Pearl (1995)'s celebrated *do-calculus* to the context of ancestral graphs. In the second result, we focus on a key component of Pearl's calculus—the property of *invariance under interventions*, and give stronger graphical conditions for this property than those implied by the first result. The second result also improves the earlier, similar results due to Spirtes et al. (1993).

Keywords: ancestral graphs, causal Bayesian network, do-calculus, intervention

1. Introduction

Intellectual curiosity aside, an important reason for people to care about causality or causal explanation is the need—for example, in policy assessment or decision making—to predict consequences of actions or interventions before actually carrying them out. Sometimes we can base that prediction on similar past interventions or experiments, in which case the inference is but an instance of the classical inductive generalization. Other times, however, we do not have access to sufficient controlled experimental studies for various reasons, and can only make passive observations before interventions take place. Under the latter circumstances, we need to reason from pre-intervention or observational data to a post-intervention setting.

A prominent machinery for causal reasoning of this kind is known as *causal Bayesian network* (Spirtes et al., 1993; Pearl, 2000), which we will describe in more detail in the next section. In this framework, once the causal structure—represented by a directed acyclic graph (DAG) over a set of attributes or random variables—is fully given, every query about post-intervention probability can be answered in terms of pre-intervention probabilities. So, if every variable in the causal structure is (passively) observed, the observational data can be used to estimate the post-intervention probability of interest.

Complications come in at least two ways. First, some variables in the causal DAG may be unobserved, or worse, unobservable. So even if the causal DAG (with latent variables) is fully known, we may not be able to predict certain intervention effects because we only have data from the marginal distribution over the observed variables instead of the joint distribution over all causally relevant variables. The question is what post-intervention probability is or is not identifiable given a causal DAG with latent variables. Much of Pearl's work (Pearl, 1995, 1998, 2000), and more recently Tian and Pearl (2004) are paradigmatic attempts to address this problem.

Second, the causal structure is seldom, if ever, fully known. In the situation we are concerned with in this paper, where no substantial background knowledge or controlled study is available, we have to rely upon observational data to inform us about causal structure. The familiar curse is that very rarely can observational data determine a unique causal structure, and many causal discovery algorithms in the literature output an equivalence class of causal structures based on observational data (Spirtes et al., 1993; Meek, 1995a; Spirtes et al., 1999; Chickering, 2002).[1] Different causal structures in the class may or may not give the same answer to a query about post-intervention probability. For a simple illustration, consider two causal Bayesian networks (see Section 2 below), $X \rightarrow Y \rightarrow Z$ and $X \leftarrow Y \rightarrow Z$, over three variables X, Y and Z. The two causal structures are indistinguishable (without strong parametric assumptions) by observational data. Suppose we are interested in the post-intervention probability distribution of Y given that X is manipulated to take some fixed value x. The structure $X \rightarrow Y \rightarrow Z$ entails that the post-intervention distribution of Y is identical to the pre-intervention distribution of Y *conditional on* $X = x$, whereas the structure $X \leftarrow Y \rightarrow Z$ entails that the post-intervention distribution of Y is identical to the pre-intervention marginal distribution of Y. So the two structures give different answers to this particular query. By contrast, if we are interested in the post-intervention distribution of Z under an intervention on Y, the two structures give the same answer.

The matter becomes formidably involved when both complications are present. Suppose we observe a set of random variables $\mathbf{O}$, but for all we know, the underlying causal structure may involve extra latent variables. We will not worry about the estimation of the pre-intervention distribution of $\mathbf{O}$ in this paper, so we may well assume for simplicity that the pre-intervention distribution of $\mathbf{O}$ is known. But we are interested in queries about post-intervention probability, such as the probability of $\mathbf{Y}$ conditional on $\mathbf{Z}$ that would result under an intervention on $\mathbf{X}$ (where $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$). The question is whether and how we can answer such queries from the given pre-intervention distribution of $\mathbf{O}$.

This problem is naturally divided into two parts. The first part is what some causal discovery algorithms attempt to achieve, namely, to learn something about the causal structure—usually features shared by all causal structures in an equivalence class—from the pre-intervention distribution of $\mathbf{O}$. The second part is to figure out, given the learned causal information, whether a post-intervention probability is identifiable in terms of pre-intervention probabilities.

This paper provides some results concerning the second part, assuming the available causal information is summarized in a (partial) *ancestral graph*. Ancestral graphical models (Richardson and Spirtes, 2002, 2003) have proved to be an elegant and useful surrogate for DAG models with latent variables (more details follow in Section 3), not the least because provably correct algorithms are available for learning an equivalence class

1. The recent work on linear non-Gaussian structural equation models (Shimizu et al., 2006) is an exception. However, we do not make parametric assumptions in this paper.

of ancestral graphs represented by a partial ancestral graph from the pre-intervention distribution of the observed variables—in particular, from the conditional independence and dependence relations implied by the distribution (Spirtes et al., 1999; Zhang, forthcoming).

We have two main results. First, we extend the *do*-calculus of Pearl (1995) to the context of ancestral graphs (Section 4), so that the resulting calculus is based on an equivalence class of causal DAGs with latent variables rather than a single one. Second, we focus on a key component of Pearl's calculus—the property of *invariance under interventions* studied by Spirtes et al. (1993), and give stronger graphical conditions for this property than those implied by the first result (Section 5). Our result improves upon the Spirtes-Glymour-Scheines conditions for invariance formulated with respect to the so-called *inducing path graphs*, whose relationship with ancestral graphs is discussed in Appendix A.

2. Causal Bayesian Network

A Bayesian network for a set of random variables $\mathbf{V}$ consists of a pair $\langle \mathcal{G}, P \rangle$, where $\mathcal{G}$ is a directed acyclic graph (DAG) with $\mathbf{V}$ as the set of vertices, and P is the joint probability function of $\mathbf{V}$, such that P factorizes according to $\mathcal{G}$ as follows:

$$P(\mathbf{V}) = \prod_{Y \in \mathbf{V}} P(Y \mid \mathbf{Pa}_{\mathcal{G}}(Y))$$

where $\mathbf{Pa}_{\mathcal{G}}(Y)$ denotes the set of parents of Y in $\mathcal{G}$. In a causal Bayesian network, the DAG $\mathcal{G}$ is interpreted causally, as a representation of the causal structure over $\mathbf{V}$. That is, for $X, Y \in \mathbf{V}$, an arrow from X to Y ($X \to Y$) in $\mathcal{G}$ means that X has a *direct* causal influence on Y relative to $\mathbf{V}$. We refer to a causally interpreted DAG as a **causal DAG**. The postulate that the (pre-intervention) joint distribution P factorizes according to the causal DAG $\mathcal{G}$ is known as the **causal Markov condition**.

What about interventions? For simplicity, let us focus on what Pearl (2000) calls *atomic* interventions—interventions that fix the values of the target variables—though the results in Section 5 also apply to more general types of interventions (such as interventions that confer a non-degenerate probability distribution on the target variables). In the framework of causal Bayesian network, an intervention on $\mathbf{X} \subseteq \mathbf{V}$ is supposed to be *effective* in the sense that the value of $\mathbf{X}$ is completely determined by the intervention, and *local* in the sense that the conditional distributions of other variables (variables not in $\mathbf{X}$) given their respective parents in the causal DAG are not affected by the intervention. Graphically, such an intervention amounts to erasing all arrows into $\mathbf{X}$ in the causal DAG (because variables in $\mathbf{X}$ do not depend on their original parents any more), but otherwise keeping the graph as it is. Call this modified graph the **post-intervention causal graph**.

Based on this understanding of interventions, the following postulate has been proposed by several authors in various forms (Robins, 1986; Spirtes et al., 1993; Pearl, 2000):

> **Intervention Principle** Given a causal DAG $\mathcal{G}$ over $\mathbf{V}$ and a (pre-intervention) joint distribution P that factorizes according to $\mathcal{G}$, the post-intervention distribution $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{V})$—that is, the joint distribution of $\mathbf{V}$ after $\mathbf{X} \subseteq \mathbf{V}$ are manipulated to values $\mathbf{x}$ by an intervention—takes a similar, truncated form of factorization, as

follows:

$$P_{\mathbf{X}:=\mathbf{x}}(\mathbf{V}) = \begin{cases} \prod_{Y \in \mathbf{V} \setminus \mathbf{X}} P(Y \mid \mathbf{Pa}_{\mathcal{G}}(Y)) & \text{for values of } \mathbf{V} \text{ consistent with } \mathbf{X} = \mathbf{x}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that in the case of a null intervention (when $\mathbf{X} = \emptyset$), the intervention principle implies the factorization of the pre-intervention distribution P according to $\mathcal{G}$, which is just the causal Markov condition. So the intervention principle generalizes the causal Markov condition: it assumes that the post-intervention distribution also satisfies the causal Markov condition with the post-intervention causal graph.

By the intervention principle, once the causal DAG is given, the post-intervention joint distribution can be calculated in terms of pre-intervention probabilities.[2] So if every variable is observed, and hence those pre-intervention probabilities can be estimated, any post-intervention probability is estimable as well.

It is time to recall the two complications mentioned in the last section. First, the intervention principle is only plausible when the given set of variables is *causally sufficient*. Here is what causal sufficiency means. Given a set of variables $\mathbf{V}$, and two variables $A, B \in \mathbf{V}$, a variable C (not necessarily included in $\mathbf{V}$) is called a *common direct cause* of A and B relative to $\mathbf{V}$ if C has a direct causal influence on A and also a direct causal influence on B relative to $\mathbf{V} \cup \{C\}$. $\mathbf{V}$ is said to be *causally sufficient* if for every pair of variables $V_1, V_2 \in \mathbf{V}$, every common direct cause of V_1 and V_2 relative to $\mathbf{V}$ is also a member of $\mathbf{V}$. It is well known that the causal Markov condition tends to fail for a causally insufficient set of variables (Spirtes et al., 1993), and even more so with the intervention principle. But in most real situations, there is no reason to assume that the set of observed variables is causally sufficient, so the causal Bayesian network may well involve latent variables.

Second, the causal DAG is not fully learnable with observational, pre-intervention data. The causal discovery algorithms in the literature—some of which are provably correct in the large sample limit assuming the causal Markov condition and its converse, causal Faithfulness condition—typically return an equivalence class of DAGs that imply the same conditional independence relations among the observed variables (according to the Markov condition), with some causal features in common that constitute the learned causal information. Given such limited causal information, a post-intervention probability may or may not be uniquely identifiable.

Taking both complications into account, the interesting question is this: what causal reasoning is warranted given the causal information learnable by algorithms that do not assume causal sufficiency for the set of observed variables, such as the FCI algorithm presented in Spirtes et al. (1999)? Before we explore the question, let us make it a little more precise with the formalism of ancestral graphs.

3. Ancestral Graphical Models

Ancestral graphical models are motivated by the need to represent data generating processes that may involve latent confounders and/or selection bias,[3] without explicitly

2. A technical issue is that some conditional probabilities may be undefined in the pre-intervention distribution. In this paper we ignore that issue by assuming that the pre-intervention distribution is strictly positive. Otherwise we just need to add the proviso "when all the conditional probabilities involved are defined" to all our results.
3. Roughly speaking, there is selection bias if the probability of a unit being sampled depends on certain properties of the unit. The kind of selection bias that is especially troublesome for causal inference is when

modelling the unobserved variables (Richardson and Spirtes, 2002). We do not deal with selection bias in this paper, so we use only part of the machinery.

A (directed) *mixed graph* is a vertex-edge graph that may contain two kinds of edges: directed edges ($\rightarrow$) and bi-directed edges ($\leftrightarrow$). Between any two vertices there is at most one edge. The two ends of an edge we call *marks*. Obviously there are two kinds of marks: *arrowhead* ($>$) and *tail* ($-$). The marks of a bi-directed edge are both arrowheads, and a directed edge has one arrowhead and one tail. We say an edge is *into* (or *out of*) a vertex if the mark of the edge at the vertex is an arrowhead (or tail).

Two vertices are said to be *adjacent* in a graph if there is an edge (of any kind) between them. Given a mixed graph $\mathcal{G}$ and two adjacent vertices X, Y therein, X is called a *parent* of Y and Y a *child* of X if $X \rightarrow Y$ is in $\mathcal{G}$; X is called a *spouse* of Y (and Y a spouse of X) if $X \leftrightarrow Y$ is in $\mathcal{G}$. A *path* in $\mathcal{G}$ is a sequence of distinct vertices $\langle V_0, ..., V_n \rangle$ such that for all $0 \leq i \leq n-1$, V_i and V_{i+1} are adjacent in $\mathcal{G}$. A *directed path from V_0 to V_n* in $\mathcal{G}$ is a sequence of distinct vertices $\langle V_0, ..., V_n \rangle$ such that for all $0 \leq i \leq n-1$, V_i is a parent of V_{i+1} in $\mathcal{G}$. X is called an *ancestor* of Y and Y a *descendant* of X if $X = Y$ or there is a directed path from X to Y. We use $\mathbf{Pa}_{\mathcal{G}}, \mathbf{Ch}_{\mathcal{G}}, \mathbf{Sp}_{\mathcal{G}}, \mathbf{An}_{\mathcal{G}}, \mathbf{De}_{\mathcal{G}}$ to denote the set of parents, children, spouses, ancestors, and descendants of a vertex in $\mathcal{G}$, respectively. A *directed cycle* occurs in $\mathcal{G}$ when $Y \rightarrow X$ is in $\mathcal{G}$ and $X \in \mathbf{An}_{\mathcal{G}}(Y)$. An *almost directed cycle* occurs when $Y \leftrightarrow X$ is in $\mathcal{G}$ and $X \in \mathbf{An}_{\mathcal{G}}(Y)$.[4]

Given a path $p = \langle V_0, \ldots, V_n \rangle$ with $n > 1$, V_i ($1 \leq i \leq n-1$) is a *collider* on p if the two edges incident to V_i are both into V_i, that is, have an arrowhead at V_i; otherwise it is called a *noncollider* on p. In Figure 1(a), for example, B is a collider on the path $\langle A, B, D \rangle$, but is a non-collider on the path $\langle C, B, D \rangle$. A *collider path* is a path on which every vertex except for the endpoints is a collider. For example, in Figure 1(a), the path $\langle C, A, B, D \rangle$ is a collider path because both A and B are colliders on the path. Let $\mathbf{L}$ be any subset of vertices in the graph. An *inducing path relative to* $\mathbf{L}$ is a path on which every vertex not in $\mathbf{L}$ (except for the endpoints) is a collider on the path and every collider is an ancestor of an endpoint of the path. For example, any single-edge path is trivially an inducing path relative to any set of vertices (because the definition does not constrain the endpoints of the path). In Figure 1(a), the path $\langle C, B, D \rangle$ is an inducing path relative to $\{B\}$, but not an inducing path relative to the empty set (because B is not a collider). However, the path $\langle C, A, B, D \rangle$ is an inducing path relative to the empty set, because both A and B are colliders on the path, A is an ancestor of D, and B is an ancestor of C. To simplify terminology, we will henceforth refer to inducing paths relative to the empty set simply as inducing paths.[5]

Definition 1 (MAG) *A mixed graph is called a maximal ancestral graph (MAG) if*

i. the graph does not contain any directed or almost directed cycles (ancestral); and

ii. there is no inducing path between any two non-adjacent vertices (maximal).

The first condition is obviously an extension of the defining condition for DAGs. It follows that in an ancestral graph an arrowhead, whether on a directed edge or a bi-directed edge, implies non-ancestorship. The second condition is a technical one, but the original motivation is the familiar pairwise Markov property of DAGs: if two vertices

two or more properties of interest affect the probability of being sampled, giving rise to "misleading" associations in the sample.

4. The terminology of "almost directed cycle" is motivated by the fact that removing the arrowhead at Y on $Y \leftrightarrow X$ results in a directed cycle.
5. They are called *primitive inducing paths* by Richardson and Spirtes (2002).

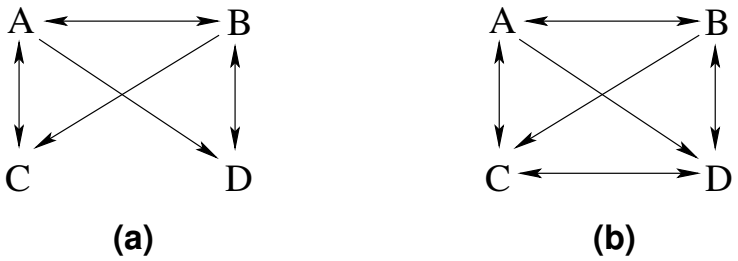


Figure 1: (a) an ancestral graph that is not maximal; (b) a maximal ancestral graph.

are not adjacent, then they are d-separated by some set of other vertices. The notion of d-separation carries over to mixed graphs in a straightforward way, as we will see shortly. But in general an ancestral graph does not need to satisfy the pairwise Markov property, or what is called maximality here. A sufficient and necessary condition for maximality turns out to be precisely the second clause in the above definition, as proved by Richardson and Spirtes (2002). So although the graph in Figure 1(a) is ancestral, it is not maximal because there is an inducing path between C and D (i.e., $\langle C, A, B, D \rangle$), but C and D are not adjacent. However, each non-maximal ancestral graph has a unique supergraph that is ancestral and maximal. For example, Figure 1(b) is the unique MAG that is also a supergraph of Figure 1(a); the former has an extra bi-directed edge between C and D.

It is worth noting that both conditions in Definition 1 are obviously met by a DAG. Hence, syntactically a DAG is also a MAG, one without bi-directed edges.

An important notion in directed graphical models is that of d-separation, which captures exactly the conditional independence relations entailed by a DAG according to the Markov condition. It is straightforward to extend the notion to mixed graphs, which, following Richardson and Spirtes (2002), we call *m-separation*.

Definition 2 (m-separation) *In a mixed graph, a path p between vertices X and Y is active (or m-connecting) relative to a (possibly empty) set of vertices $\mathbf{Z}$ ($X, Y \notin \mathbf{Z}$) if*

i. every non-collider on p is not a member of $\mathbf{Z}$;

ii. every collider on p is an ancestor of some member of $\mathbf{Z}$.

X and Y are said to be m-separated by $\mathbf{Z}$ if there is no active path between X and Y relative to $\mathbf{Z}$.

Two disjoint sets of variables $\mathbf{X}$ and $\mathbf{Y}$ are m-separated by $\mathbf{Z}$ if every variable in $\mathbf{X}$ is m-separated from every variable in $\mathbf{Y}$ by $\mathbf{Z}$.

In DAGs, obviously, m-separation reduces to d-separation. The (global) Markov property of ancestral graphical models is defined by m-separation.

A nice property of MAGs is that they can represent the marginal independence models of DAGs in the following sense: given any DAG $\mathcal{G}$ over $\mathbf{V} = \mathbf{O} \cup \mathbf{L}$—where $\mathbf{O}$ denotes the set of observed variables, and $\mathbf{L}$ denotes the set of latent variables—there is a MAG over $\mathbf{O}$ alone such that for any disjoint $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $\mathcal{G}$ (and hence entailed by $\mathcal{G}$ to be independent conditional on $\mathbf{Z}$) if and only if they are m-separated by $\mathbf{Z}$ in the MAG (and hence entailed by the MAG to be independent conditional on $\mathbf{Z}$). The following construction gives us such a MAG:

Input: a DAG $\mathcal{G}$ over $\langle \mathbf{O}, \mathbf{L} \rangle$
Output: a MAG $\mathcal{M}_\mathcal{G}$ over $\mathbf{O}$

1. for each pair of variables $A, B \in \mathbf{O}$, A and B are adjacent in $\mathcal{M}_\mathcal{G}$ if and only if there is an inducing path between them relative to $\mathbf{L}$ in $\mathcal{G}$;

2. for each pair of adjacent variables A, B in $\mathcal{M}_\mathcal{G}$, orient the edge as $A \rightarrow B$ in $\mathcal{M}_\mathcal{G}$ if A is an ancestor of B in $\mathcal{G}$; orient it as $A \leftarrow B$ in $\mathcal{M}_\mathcal{G}$ if B is an ancestor of A in $\mathcal{G}$; orient it as $A \leftrightarrow B$ in $\mathcal{M}_\mathcal{G}$ otherwise.

It can be shown that $\mathcal{M}_\mathcal{G}$ is indeed a MAG and represents the marginal independence model over $\mathbf{O}$ (Richardson and Spirtes (2002); also see Lemma 20 below). More importantly, $\mathcal{M}_\mathcal{G}$ also retains the ancestral relationships—and hence causal relationships under the standard interpretation—among $\mathbf{O}$. So, if $\mathcal{G}$ is the causal DAG for $\langle \mathbf{O}, \mathbf{L} \rangle$, it is fair to call $\mathcal{M}_\mathcal{G}$ the **causal MAG** for $\mathbf{O}$. Henceforth when we speak of a MAG over $\mathbf{O}$ *representing* a DAG over $\langle \mathbf{O}, \mathbf{L} \rangle$, we mean that the MAG is the output of the above construction procedure applied to the DAG.

Different causal DAGs may correspond to the same causal MAG. So essentially a MAG represents a set of DAGs that have the exact same d-separation structures and ancestral relationships among the observed variables. A causal MAG thus carries uncertainty about what the true causal DAG is, but also reveals features that must be satisfied by the underlying causal DAG.

There is then a natural causal interpretation of the edges in MAGs, derivative from the causal interpretation of DAGs. A directed edge from A to B in a MAG means that A is a cause of B (which is a shorthand way of saying that there is a causal pathway from A to B in the underlying DAG); a bi-directed edge between A and B means that A is not a cause of B and B is not a cause of A, which implies that there is a latent common cause of A and B (i.e., there is a latent variable L in the underlying DAG such that there is a directed path from L to A and a directed path from L to B[6]).

We borrow a simple example from Spirtes et al. (1993) to illustrate various concepts and results in this paper. Suppose we are able to observe the following variables: *Income* (I), *Parents' smoking habits* (PSH), *Smoking* (S), *Genotype* (G) and *Lung cancer* (L). The data, for all we know, are generated according to an underlying mechanism which might involve unobserved common causes. Suppose, unknown to us, the structure of the causal mechanism is the one in Figure 2, where *Profession* is an unmeasured common cause of *Income* and *Smoking*.[7]

The causal MAG that corresponds to the causal DAG is depicted in Figure 3(a)—which *syntactically* happens to be a DAG in this case. This MAG can represent some other DAGs as well. For example, it can also represent the DAG with an extra latent common cause of PSH and S.

In general a MAG is still not fully testable with observational data. Just as different DAGs can share the exact same d-separation features and hence entail the exact same conditional independence constraints, different MAGs can entail the exact same constraints by the m-separation criterion. This is known as *Markov equivalence*. Several characterizations of the Markov equivalence between MAGs are available (Spirtes and Richardson, 1996; Ali et al., 2004; Zhang and Spirtes, 2005; Zhao et al., 2005). For the

6. Note that a latent common cause is not necessarily a common *direct* cause as defined on page 76. The path from L to A, for example, may include other observed variables.
7. This example is used purely for illustrative purposes, so we will not worry why Profession is not observed but Genotype is. The exact domains of the variables do not matter either.

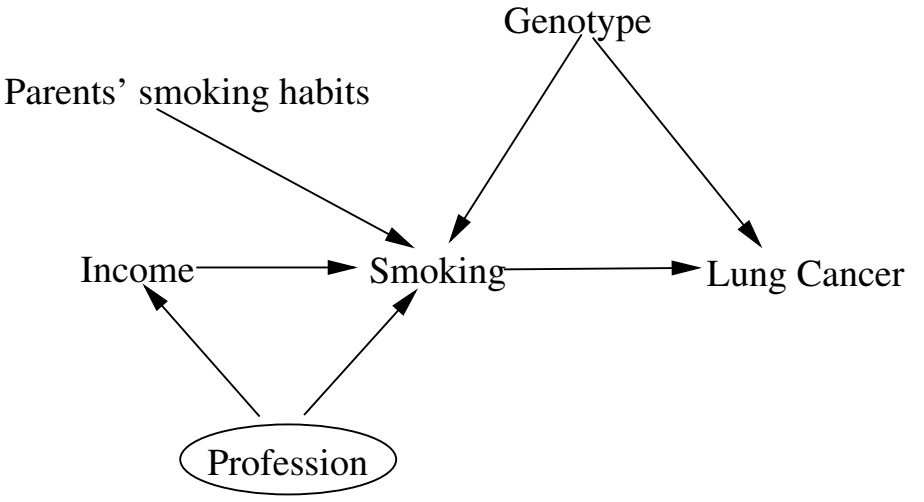


Figure 2: A causal DAG with a latent variable.

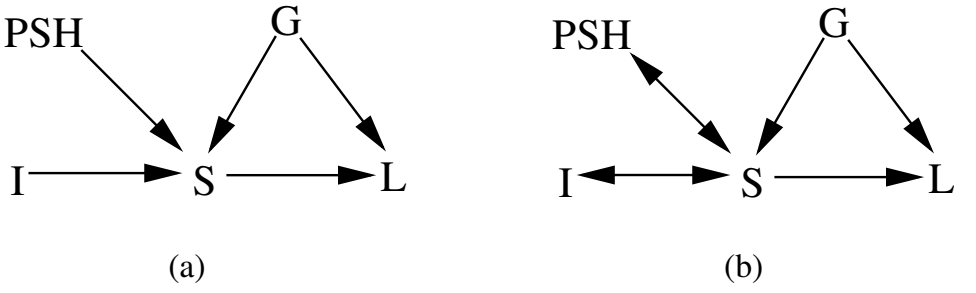


Figure 3: Two Markov Equivalent MAGs.

purpose of the present paper, it suffices to note that, as is the case with DAGs, all Markov equivalent MAGs have the same adjacencies and usually some common edge orientations as well. For example, the two MAGs in Figure 3 are Markov equivalent.

This motivates the following representation of equivalence classes of MAGs. Let *partial mixed graphs* denote the class of graphs that can contain four kinds of edges: $\rightarrow$, $\leftrightarrow$, $\circ\!\!-\!\!\circ$ and $\circ\!\!\rightarrow$, and hence three kinds of end marks for edges: arrowhead ($>$), tail ($-$) and circle ($\circ$).

Definition 3 (PAG) *Let $[\mathcal{M}]$ be the Markov equivalence class of an arbitrary MAG $\mathcal{M}$. The partial ancestral graph (PAG) for $[\mathcal{M}]$, $\mathcal{P}_{[\mathcal{M}]}$, is a partial mixed graph such that*

i. $\mathcal{P}_{[\mathcal{M}]}$ has the same adjacencies as $\mathcal{M}$ (and any member of $[\mathcal{M}]$) does;

ii. A mark of arrowhead is in $\mathcal{P}_{[\mathcal{M}]}$ if and only if it is shared by all MAGs in $[\mathcal{M}]$; and

iii. A mark of tail is in $\mathcal{P}_{[\mathcal{M}]}$ if and only if it is shared by all MAGs in $[\mathcal{M}]$.[8]

Basically a PAG represents an equivalence class of MAGs by displaying all common edge marks shared by all members in the class and displaying circles for those marks that are not common, much in the same way that a so-called Pattern (a.k.a. a PDAG or an essential graph) represents an equivalence class of DAGs (see, e.g., Spirtes et al. (1993, chap. 5); Chickering (1995); Andersson et al. (1997). For instance, the PAG for our running example is drawn in Figure 4, which displays all the commonalities among MAGs that are Markov equivalent to the MAGs in Figure 3.

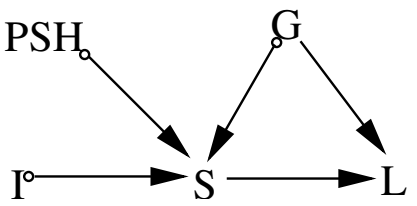


Figure 4: The PAG in our five-variable example.

Different PAGs, representing different equivalence classes of MAGs, entail different sets of conditional independence constraints. Hence a PAG is in principle fully testable by the conditional independence relations among the observed variables. Assuming the causal Markov condition and its converse, the causal Faithfulness condition,[9] there is a provably correct independence-constraint-based algorithm to learn a PAG from

8. This defines what Zhang (2006, pp. 71) calls *complete* or *maximally oriented* PAGs. In this paper, we do not consider PAGs that fail to display all common edge marks in an equivalence class of MAGs (as, e.g., allowed in Spirtes et al., 1999), so we will simply use 'PAG' to mean 'maximally oriented PAG'.

9. We have introduced the causal Markov condition in its factorization form. In terms of d-separation, the causal Markov condition says that d-separation in a causal DAG implies conditional independence in the (pre-intervention) population distribution. The causal Faithfulness condition says that d-connection in a causal DAG implies conditional dependence in the (pre-intervention) population distribution. Given the exact correspondence between d-separation relations among the observed variables in the causal DAG and m-separation relations in the causal MAG, the two conditions imply that conditional independence relations among the observed variables correspond exactly to m-separation in the causal MAG, which forms the basis of constraint-based learning algorithms.

an oracle of conditional independence relations (Spirtes et al. (1999); Zhang (2006, chap. 3)).[10] Score-based algorithms for learning PAGs are also under investigation.

Directed paths and ancestors/descendants in a PAG are defined in the same way as in a MAG. In addition, a path between X and Y, $\langle X = V_0, \ldots, V_n = Y \rangle$, is called a *possibly directed path* from X to Y[11] if for every $0 < i \leq n$, the edge between V_{i-1} and V_i is not into V_{i-1}. Call X a *possible ancestor* of Y (and Y a *possible descendant* of X) if $X = Y$ or there is a possibly directed path from X to Y in the PAG.[12] For example, in Figure 4, the path $\langle I, S, L \rangle$ is a possibly directed path, and I is a possible ancestor of L. We use $\mathbf{PossibleAn}_{\mathcal{P}}(Y)$ to denote the set of possible ancestors of Y in $\mathcal{P}$.

In partial mixed graphs two analogues of m-connecting paths will play a role later. Let p be any path in a partial mixed graph, and W be any (non-endpoint) vertex on p. Let U and V be the two vertices adjacent to W on p. W is a *collider* on p if, as before, both the edge between U and W and the edge between V and W are into W (i.e., have an arrowhead at W, $U *\!\!\rightarrow W \leftarrow\!\!* V$). W is called a *definite non-collider* on p if the edge between U and W or the edge between V and W is out of W (i.e., has a tail at W, $U \leftarrow W *\!\!-\!\!* V$ or $U *\!\!-\!\!* W \rightarrow V$), or both edges have a circle mark at W and there is no edge between U and V (i.e., $U *\!\!-\!\!\circ W \circ\!\!-\!\!* V$, where U and V are not adjacent).[13] The first analogue of m-connecting path is the following:

Definition 4 (Definite m-connecting path) *In a partial mixed graph, a path p between two vertices X and Y is a definite m-connecting path relative to a (possibly empty) set of vertices $\mathbf{Z}$ ($X, Y \notin \mathbf{Z}$) if every non-endpoint vertex on p is either a definite non-collider or a collider and*

i. every definite non-collider on p is not a member of $\mathbf{Z}$;

ii. every collider on p is an ancestor of some member of $\mathbf{Z}$.

It is not hard to see that if there is a definite m-connecting path between X and Y given $\mathbf{Z}$ in a PAG, then in every MAG represented by the PAG, the corresponding path is an m-connecting path between X and Y given $\mathbf{Z}$. For example, in Figure 4 the path $\langle I, S, G \rangle$ is definitely m-connecting given L, and this path is m-connecting given L in every member of the equivalence class. A quite surprising result is that if there is an m-connecting path between X and Y given $\mathbf{Z}$ in a MAG, then there must be a definite m-connecting path (not necessarily the same path) between X and Y given $\mathbf{Z}$ in its PAG, which we will use in Section 5.

Another analogue of m-connecting path is the following:

10. It is essentially the FCI algorithm (Spirtes et al., 1999), but with slight modifications (Zhang, 2006, chap. 3). The implemented FCI algorithm in the Tetrad IV package (`http://www.phil.cmu.edu/projects/tetrad/tetrad4.html`) is the modified version. By the way, if we also take into account the possibility of selection bias, then we need to consider a broader class of MAGs which can contain undirected edges, and the FCI algorithm needs to be augmented with additional edge inference rules (Zhang, 2006, chap. 4; forthcoming).
11. It is named a *potentially directed path* in Zhang (2006, pp. 99). The present terminology is more consistent with the names for other related notions, such as possible ancestor, possibly m-connecting path, etc.
12. The qualifier 'possible/possibly' is used to indicate that there is some MAG represented by the PAG in which the corresponding path is directed, and X is an ancestor of Y. This is not hard to establish given the valid procedure for constructing representative MAGs from a PAG presented in Lemma 4.3.6 of Zhang (2006) or Theorem 2 of Zhang (forthcoming).
13. '*' is used as wildcard that denotes any of the three possible marks: circle, arrowhead, and tail. When the graph is a PAG for some equivalence class of MAGs, the qualifier 'definite' is used to indicate that the vertex is a non-collider on the path in each and every MAG represented by the PAG, even though the circles may correspond to different marks in different MAGs. The reason why $U *\!\!-\!\!\circ W \circ\!\!-\!\!* V$ is a definite non-collider when U and V are not adjacent is because if it were a collider, it would be shared by all Markov equivalent MAGs, and hence would be manifest in the PAG.

Definition 5 (Possibly m-connecting path) *In a partial mixed graph, a path p between vertices X and Y is possibly m-connecting relative to a (possibly empty) set of vertices* $\mathbf{Z}$ *($X, Y \notin \mathbf{Z}$) if*

i. every definite non-collider on p is not a member of $\mathbf{Z}$*;*

ii. every collider on p is a possible ancestor of some member of $\mathbf{Z}$*.*

Obviously a definite m-connecting path is also a possibly m-connecting path, but not necessarily vice versa. In particular, on a possibly m-connecting path it is not required that every (non-endpoint) vertex be of a "definite" status. Figure 5 provides an illustration. The graph on the right is the PAG for the equivalence class that contains the MAG on the left (in this case, unfortunately, no informative edge mark is revealed in the PAG). In the PAG, the path $\langle X, Y, Z, W \rangle$ is a possibly m-connecting path but not a definite m-connecting path relative to $\{Y, Z\}$, because Y and Z are neither colliders nor definite non-colliders on the path. Note that in the MAG, $\langle X, Y, Z, W \rangle$ is not m-connecting relative to $\{Y, Z\}$. In fact, X and W are m-separated by $\{Y, Z\}$ in the MAG. So unlike a definite m-connecting path, a mere possibly m-connecting path in a PAG does not necessarily correspond to a m-connecting path (or imply the existence of a m-connecting path) in a representative MAG in the equivalence class.[14]

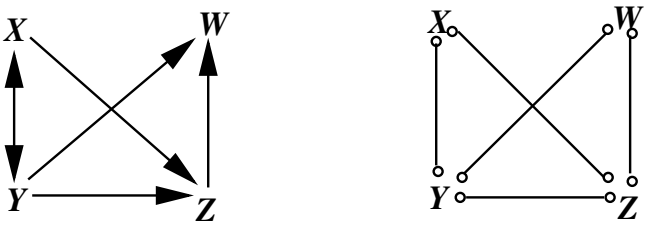


Figure 5: Difference between possible and definite m-connecting paths: in the PAG on the right, $\langle X, Y, Z, W \rangle$ is a possibly m-connecting path relative to $\{Y, Z\}$ but *not* a definite m-connecting path relative to $\{Y, Z\}$. Also note that $\langle X, Y, Z, W \rangle$ is *not* m-connecting relative to $\{Y, Z\}$ in the MAG on the left, even though the MAG is a member of the equivalence class represented by the PAG.

As we will see, the main result in Section 4 is formulated in terms of absence of possibly m-connecting paths (what we will call, for want of a better term, definite m-separation), whereas the main result in Section 5 is formulated in terms of absence of definite m-connecting paths. This is one important aspect in which the result in Section 5 is better than that in Section 4 (and than the analogous results presented in Spirtes et al. (1993)) regarding the property of invariance under interventions. We will come back to this point after we present the PAG-based *do*-calculus.

14. This case is even more extreme in that in *every* MAG that belongs to the equivalence class, X and W are m-separated by Y and Z. So this example can be used to show that the *do*-calculus developed in Section 4 is not yet complete, though it is not clear how serious the incompleteness is.

4. Do-Calculus

Pearl (1995) developed an elegant *do*-calculus for identifying post-intervention probabilities given a single causal DAG with (or without) latent variables. To honor the name of the calculus, in this section we will use Pearl's *'do' operator* to denote post-intervention probabilities. Basically, the notation we used for the post-intervention probability function under an intervention on $\mathbf{X}$, $P_{\mathbf{X}:=\mathbf{x}}(\bullet)$, will be written as $P(\bullet \mid do(\mathbf{X} = \mathbf{x}))$.

The calculus contains three inference rules whose antecedents make reference to surgeries on the given causal DAG. There are two types of graph manipulations:

Definition 6 (Manipulations of DAGs) *Given a DAG $\mathcal{G}$ and a set of variables $\mathbf{X}$ therein,*

- *the* ***X-lower-manipulation*** *of $\mathcal{G}$ deletes all edges in $\mathcal{G}$ that are out of variables in $\mathbf{X}$, and otherwise keeps $\mathcal{G}$ as it is. The resulting graph is denoted as $\mathcal{G}_{\underline{\mathbf{X}}}$.*
- *the* ***X-upper-manipulation*** *of $\mathcal{G}$ deletes all edges in $\mathcal{G}$ that are into variables in $\mathbf{X}$, and otherwise keeps $\mathcal{G}$ as it is. The resulting graph is denoted as $\mathcal{G}_{\overline{\mathbf{X}}}$.*

The following proposition summarizes Pearl's *do*-calculus. (Following Pearl, we use lower case letters to denote generic value settings for the sets of variables denoted by the corresponding upper case letters. So for simplicity we write $P(\mathbf{x})$ to mean $P(\mathbf{X} = \mathbf{x})$, and $do(\mathbf{x})$ to mean $do(\mathbf{X} = \mathbf{x})$.)

Proposition 7 (Pearl) *Let $\mathcal{G}$ be the causal DAG for $\mathbf{V}$, and $\mathbf{U}, \mathbf{X}, \mathbf{Y}, \mathbf{W}$ be disjoint subsets of $\mathbf{V}$. The following rules are sound:*

1. *if $\mathbf{Y}$ and $\mathbf{X}$ are d-separated by $\mathbf{U} \cup \mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{U}}}$, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w}).$$

2. *if $\mathbf{Y}$ and $\mathbf{X}$ are d-separated by $\mathbf{U} \cup \mathbf{W}$ in $\mathcal{G}_{\underline{\mathbf{X}}\overline{\mathbf{U}}}$, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}).$$

3. *if $\mathbf{Y}$ and $\mathbf{X}$ are d-separated by $\mathbf{U} \cup \mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{U}}\overline{\mathbf{X}'}}$, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w})$$

where $\mathbf{X}' = \mathbf{X} \backslash \mathbf{An}_{\mathcal{G}_{\overline{\mathbf{U}}}}(\mathbf{W}) = \mathbf{X} \backslash (\cup_{W \in \mathbf{W}} \mathbf{An}_{\mathcal{G}_{\overline{\mathbf{U}}}}(W))$.

The proposition follows from the intervention principle (Pearl, 1995). The first rule is actually not independent—it can be derived from the other two rules (Huang and Valtorta, 2006), but it has long been an official part of the calculus. The soundness of the calculus ensures that any post-intervention probability that can be reduced via the calculus to an expression that only involves pre-intervention probabilities of observed variables is identifiable. Recently, the completeness of the calculus was also established, in the sense that any identifiable post-intervention probability can be so reduced using the calculus (Huang and Valtorta, 2006; Shpister and Pearl, 2006).

Our goal is to develop a similar calculus when the available causal information is given in a PAG. A natural idea is to formulate analogous inference rules in terms of (manipulated) PAGs, to the effect that if a certain rule is applicable given a PAG, the corresponding rule in Pearl's calculus will be applicable given the (unknown) true causal DAG. How to guarantee that? Recall that a PAG represents an equivalence class

of MAGs; each MAG, in turn, represents a set of causal DAGs. The union of all these sets is the set of DAGs represented by the PAG—one of them is the true causal DAG. So a sure way to get what we want is to formulate analogous rules in terms of PAGs such that if the rule is applicable given a PAG, then for every DAG represented by the PAG, the corresponding rule in Pearl's calculus is applicable.

For this purpose, it is natural to develop the desired calculus in two steps. First, we derive an analogous *do*-calculus based on MAGs, such that if a rule is applicable given a MAG, then for every DAG represented by the MAG, the corresponding rule in Pearl's calculus is applicable. Second, we extend that to a *do*-calculus based on PAGs, such that if a rule is applicable given a PAG, then for every MAG in the equivalence class represented by the PAG, the corresponding rule in the MAG-based calculus is applicable.

Before we define appropriate analogues of graph manipulations on MAGs, it is necessary to distinguish two kinds of directed edges in a MAG, according to the following criterion.

Definition 8 (Visibility) *Given a MAG $\mathcal{M}$, a directed edge $A \to B$ in $\mathcal{M}$ is **visible** if there is a vertex C not adjacent to B, such that either there is an edge between C and A that is into A, or there is a collider path between C and A that is into A and every vertex on the path is a parent of B. Otherwise $A \to B$ is said to be **invisible**.*

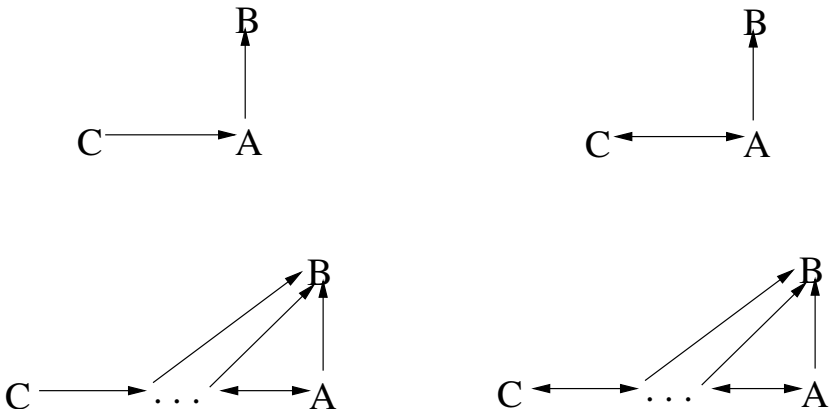


Figure 6: Possible configurations of visibility for $A \to B$.

Figure 6 gives the possible configurations that make a directed edge $A \to B$ visible. The distinction between visible and invisible directed edges is important because of the following two facts.

Lemma 9 *Let $\mathcal{G}$ be a DAG over $\mathbf{O} \cup \mathbf{L}$, and $\mathcal{M}$ be the MAG over $\mathbf{O}$ that represents the DAG. For any $A, B \in \mathbf{O}$, if $A \in \mathbf{An}_{\mathcal{G}}(B)$, and there is an inducing path relative to $\mathbf{L}$ between A and B that is into A in $\mathcal{G}$, then there is a directed edge $A \to B$ in $\mathcal{M}$ that is invisible.*

Proof See Appendix B. ■

Taking the contrapositive of Lemma 9 gives us the fact that if $A \to B$ is visible in a MAG, then in *every* DAG represented by the MAG, there is no inducing path between A

and B relative to the set of latent variables that is also into A. This implies that for every such DAG G, $G_{\underline{A}}$—the graph resulting from eliminating edges out of A in G—will not contain any inducing path between A and B relative to the set of latent variables, which means that the MAG that represents $G_{\underline{A}}$ will not contain any edge between A and B. So intuitively, deleting edges out of A in the underlying DAG corresponds to deleting visible arrows out of A in the MAG.

How about invisible arrows? Here is the relevant fact.

Lemma 10 *Let $\mathcal{M}$ be any MAG over a set of variables $\mathbf{O}$, and $A \to B$ be any directed edge in $\mathcal{M}$. If $A \to B$ is invisible in $\mathcal{M}$, then there is a DAG whose MAG is $\mathcal{M}$ in which A and B share a latent parent, that is, there is a latent variable L_{AB} in the DAG such that $A \leftarrow L_{AB} \to B$ is a subgraph of the DAG.*

Proof See Appendix B. ■

Obviously $A \leftarrow L_{AB} \to B$ is an inducing path between A and B relative to the set of latent variables. So if $A \to B$ in a MAG is invisible, at least for *some* DAG G represented by the MAG—and for all we know, this DAG may well be the true causal DAG—$G_{\underline{A}}$ contains $A \leftarrow L_{AB} \to B$, and hence corresponds to a MAG in which $A \leftrightarrow B$ appears.

Finally, for either $A \leftrightarrow B$ or $A \to B$ in a MAG, it is not hard to show that for *every* DAG represented by the MAG, there is no inducing path in the DAG between A and B relative to the set of latent variables that is also out of B (since otherwise B would be an ancestor of A, violating the definition of ancestral graphs). So deleting edges into B in the underlying DAG corresponds to deleting edges into B in the MAG. These considerations motivate the following definition.

Definition 11 (Manipulations of MAGs) *Given a MAG $\mathcal{M}$ and a set of variables $\mathbf{X}$ therein,*

- *the* **$\mathbf{X}$-*lower-manipulation*** *of $\mathcal{M}$ deletes all those edges that are visible in $\mathcal{M}$ and are out of variables in $\mathbf{X}$, replaces all those edges that are out of variables in $\mathbf{X}$ but are invisible in $\mathcal{M}$ with bi-directed edges, and otherwise keeps $\mathcal{M}$ as it is. The resulting graph is denoted as $\mathcal{M}_{\underline{\mathbf{X}}}$.*
- *the* **$\mathbf{X}$-*upper-manipulation*** *of $\mathcal{M}$ deletes all those edges in $\mathcal{M}$ that are into variables in $\mathbf{X}$, and otherwise keeps $\mathcal{M}$ as it is. The resulting graph is denoted as $\mathcal{M}_{\overline{\mathbf{X}}}$.*

We stipulate that lower-manipulation has a higher priority than upper-manipulation, so that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ (or $\mathcal{M}_{\overline{\mathbf{X}}\underline{\mathbf{Y}}}$) denotes the graph resulting from applying the $\mathbf{X}$-upper-manipulation to the $\mathbf{Y}$-lower-manipulated graph of $\mathcal{M}$.

A couple of comments are in order. First, unlike the case of DAGs, the lower-manipulation for MAGs may introduce new edges, that is, replacing invisible directed edges with bi-directed edges. Again, the reason we do this is that an invisible directed edge from A to B allows the possibility of a latent common parent of A and B in the underlying DAG. If so, the A-lower-manipulated DAG will correspond to a MAG in which there is a bi-directed edge between A and B. Second, because of the possibility of introducing new bi-directed edges, we need the priority stipulation that lower-manipulation is to be done before upper-manipulation. The stipulation is not necessary for DAGs, because no new edges would be introduced in the lower-manipulation of DAGs, and hence the order does not matter.

Ideally, if $\mathcal{M}$ is the MAG of a DAG $\mathcal{G}$, we would like $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ to be the MAG of $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. But this is not always possible, as two DAGs represented by the same MAG before a

manipulation may correspond to different MAGs after the manipulation. But we still have the following fact:

Lemma 12 *Let $\mathcal{G}$ be a DAG over $\mathbf{O} \cup \mathbf{L}$, and $\mathcal{M}$ be the MAG of $\mathcal{G}$ over $\mathbf{O}$. Let $\mathbf{X}$ and $\mathbf{Y}$ be two possibly empty subsets of $\mathbf{O}$, and $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$ be the MAG of $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain A or B, if there is an m-connecting path between A and B given $\mathbf{C}$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, then there is an m-connecting path between A and B given $\mathbf{C}$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$.*

Proof See Appendix B. ■

Recall that a graphical model is called an *independence map* of another if any independence implied by the former is also implied by the latter (Chickering, 2002). So another way of putting Lemma 12 is that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is an independence map of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, which we write as $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}} \leq \mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. The diagram in Figure 7 visualizes what is going on.

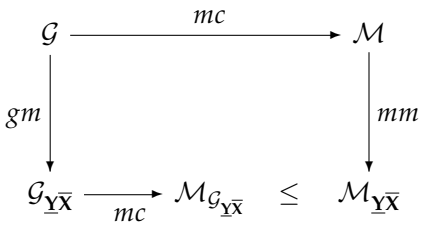


Figure 7: Illustration of Lemma 12: *mc* refers to MAG construction introduced in Section 3; *gm* refers to DAG manipulation; and *mm* refers to MAG manipulation.

Corollary 13 *Let $\mathcal{M}$ be a MAG over $\mathbf{O}$, and $\mathbf{X}$ and $\mathbf{Y}$ be two subsets of $\mathbf{O}$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain A or B, if A and B are m-separated by $\mathbf{C}$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, then A and B are d-separated by $\mathbf{C}$ in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ for every $\mathcal{G}$ represented by $\mathcal{M}$.*

Proof By Lemma 12, if A and B are m-separated by $\mathbf{C}$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, they are also m-separated by $\mathbf{C}$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, for every $\mathcal{G}$ represented by $\mathcal{M}$, which in turn implies that A and B are d-separated by $\mathbf{C}$ in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ for every $\mathcal{G}$ represented by $\mathcal{M}$, because d-separation relations among $\mathbf{O}$ in a DAG correspond exactly to m-separation relations in its MAG. ■

The converse of Corollary 13, however, is not true in general. To give the simplest example, consider the MAG $\mathcal{M}$ in Figure 8(a): $X \leftarrow Y \rightarrow Z$ (which happens to be a DAG syntactically). The two DAGs, $\mathcal{G}1$ in 8(b) and $\mathcal{G}2$ in 8(c), are both represented by $\mathcal{M}$. By the definition of lower-manipulation, $\mathcal{M}_{\underline{Y}}$ is the graph $X \leftrightarrow Y \leftrightarrow Z$. On the other hand, $\mathcal{G}1_{\underline{Y}}$ is $X \leftarrow L1 \rightarrow Y \quad Z$; and $\mathcal{G}2_{\underline{Y}}$ is $X \quad Y \leftarrow L2 \rightarrow Z$. Obviously, the MAG of $\mathcal{G}1_{\underline{Y}}$ is $X \leftrightarrow Y \quad Z$, and the MAG of $\mathcal{G}2_{\underline{Y}}$ is $X \quad Y \leftrightarrow Z$, both of which are *proper* subgraphs of $\mathcal{M}_{\underline{Y}}$. So an m-separation relation in $\mathcal{M}_{\underline{Y}}$—for example, X and Z are m-separated by the empty set—corresponds to a d-separation relation in both $\mathcal{G}1_{\underline{Y}}$ and $\mathcal{G}2_{\underline{Y}}$, in accord with Corollary 13.

By contrast, the converse of Corollary 13 fails for $\mathcal{M}$. It can be checked that for every $\mathcal{G}$ represented by $\mathcal{M}$, X and Z are d-separated by Y in $\mathcal{G}_{\underline{Y}}$, as evidenced by $\mathcal{G}1_{\underline{Y}}$ and $\mathcal{G}2_{\underline{Y}}$. But X and Z are not m-separated by Y in $\mathcal{M}_{\underline{Y}}$.

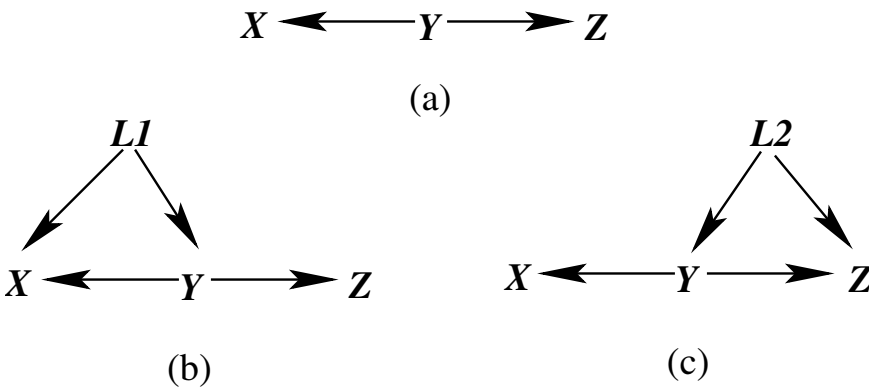


Figure 8: A counterexample to the converse of Corollary 13.

However, Definition 11 is not to be blamed for this limitation. In this simple example, one can easily enumerate all possible directed mixed graphs over X, Y, Z and see that for none of them do both Corollary 13 and its converse hold. Intuitively, this is because the MAG in Figure 8(a) implies that either $\langle X, Y\rangle$ does not have a common latent parent or $\langle Y, Z\rangle$ does not have a common latent parent in the underlying DAG. So under the Y-lower-manipulation of the underlying DAG, for all we know, either $\langle X, Y\rangle$ or $\langle Y, Z\rangle$ will become unconnected. But this disjunctive information cannot be precisely represented by a single graph.

More generally, no matter how we define $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, as long as it is a single graph, the converse of Corollary 13 will not hold in general, unless Corollary 13 itself fails. $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, as a single graph, can only aim to be a supergraph (up to Markov equivalence) of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$ for every $\mathcal{G}$ represented by $\mathcal{M}$ (which makes Corollary 13 true). To this end, Definition 11 is 'minimal' in the following sense: two variables are adjacent in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ if and only if there exists a DAG $\mathcal{G}$ represented by $\mathcal{M}$ such that the two variables are adjacent in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$. In this regard, $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ does not have more edges than necessary. One can, for example, check this fact for the simple case in Figure 8.

We are now ready to state the intermediate theorem on MAG-based *do*-calculus.

Theorem 14 (*do*-calculus given a MAG) *Let $\mathcal{M}$ be the causal MAG over $\mathbf{O}$, and $\mathbf{U}, \mathbf{X}, \mathbf{Y}, \mathbf{W}$ be disjoint subsets of $\mathbf{O}$. The following rules are valid, in the sense that if the antecedent of the rule holds, then the consequent holds no matter which DAG represented by $\mathcal{M}$ is the true causal DAG.*

1. *if $\mathbf{Y}$ and $\mathbf{X}$ are m-separated by $\mathbf{U} \cup \mathbf{W}$ in $\mathcal{M}_{\overline{\mathbf{U}}}$, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w}).$$

2. *if $\mathbf{Y}$ and $\mathbf{X}$ are m-separated by $\mathbf{U} \cup \mathbf{W}$ in $\mathcal{M}_{\underline{\mathbf{X}}\overline{\mathbf{U}}}$, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}).$$

3. *if* $\mathbf{Y}$ *and* $\mathbf{X}$ *are m-separated by* $\mathbf{U} \cup \mathbf{W}$ *in* $\mathcal{M}_{\overline{\mathbf{U}}\overline{\mathbf{X}'}}$, *then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w})$$

where $\mathbf{X}' = \mathbf{X} \backslash \mathbf{An}_{\mathcal{M}_{\overline{\mathbf{U}}}}(\mathbf{W})$.

Proof This readily follows from Proposition 7, Corollary 13, and the fact that for every $\mathcal{G}$ represented by $\mathcal{M}$, $\mathbf{An}_{\mathcal{G}_{\overline{\mathbf{U}}}}(\mathbf{W}) \cap \mathbf{O} = \mathbf{An}_{\mathcal{M}_{\overline{\mathbf{U}}}}(\mathbf{W})$. ■

As already noted, the true causal MAG is not uniquely recoverable from the pre-intervention distribution, thanks to Markov equivalence. So the main value of Theorem 14 is to facilitate the development of a PAG-based *do*-calculus. However, it is worth noting that when supplemented with some background causal knowledge, such as knowledge of the form that some variable is not a cause of another variable, it is in principle possible to determine that the true causal MAG belongs to a proper subset of the full equivalence class represented by the PAG. Depending on how strong the background knowledge is, the subset could be as big as the full equivalence class or as small as a singleton. In this sense, Theorem 14 and Theorem 17 below may be viewed as two extreme cases of a more general *do*-calculus based on a subset of Markov equivalent MAGs.

To extend the calculus to PAGs, we need to define manipulations on PAGs. They are essentially the same as the manipulations of MAGs. The definition of visibility still makes sense in PAGs, except that we will call a directed edge in a PAG *definitely visible* if it satisfies the condition for visibility in Definition 8, in order to emphasize that this edge is visible in all MAGs in the equivalence class. Despite the extreme similarity to manipulations on MAGs, let us still write down the definition of PAG manipulations for easy reference.

Definition 15 (Manipulations of PAGs) *Given a PAG* $\mathcal{P}$ *and a set of variables* $\mathbf{X}$ *therein,*

- *the* **X-*lower-manipulation*** *of* $\mathcal{P}$ *deletes all those edges that are definitely visible in* $\mathcal{P}$ *and are out of variables in* $\mathbf{X}$, *replaces all those edges that are out of variables in* $\mathbf{X}$ *but are not definitely visible in* $\mathcal{P}$ *with bi-directed edges, and otherwise keeps* $\mathcal{P}$ *as it is. The resulting graph is denoted as* $\mathcal{P}_{\underline{\mathbf{X}}}$.
- *the* **X-*upper-manipulation*** *of* $\mathcal{P}$ *deletes all those edges in* $\mathcal{P}$ *that are into variables in* $\mathbf{X}$, *and otherwise keeps* $\mathcal{P}$ *as it is. The resulting graph is denoted as* $\mathcal{P}_{\overline{\mathbf{X}}}$.

We stipulate that lower-manipulation has a higher priority than upper-manipulation, so that $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ *(or* $\mathcal{P}_{\overline{\mathbf{X}}\underline{\mathbf{Y}}}$*) denotes the graph resulting from applying the* $\mathbf{X}$*-upper-manipulation to the* $\mathbf{Y}$*-lower-manipulated graph of* $\mathcal{P}$.

We should emphasize that except in rare situations, $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is not a PAG any more (i.e., not a PAG for any Markov equivalence class of MAGs). But from $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ we still gain information about m-separation in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, where $\mathcal{M}$ is a MAG that belongs to the Markov equivalence class represented by $\mathcal{P}$. Here is a simple connection. Given a MAG $\mathcal{M}$ and the PAG $\mathcal{P}$ that represents $[\mathcal{M}]$, a trivial fact is that a m-connecting path in $\mathcal{M}$ is also a possibly m-connecting path in $\mathcal{P}$. This is also true for $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ and $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$.

Lemma 16 *Let* $\mathcal{M}$ *be a MAG over* $\mathbf{O}$, *and* $\mathcal{P}$ *be the PAG for* $[\mathcal{M}]$. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two subsets of* $\mathbf{O}$. *For any* $A, B \in \mathbf{O}$ *and* $\mathbf{C} \subseteq \mathbf{O}$ *that does not contain* A *or* B, *if a path* p *between* A *and*

B is m-connecting given $\mathbf{C}$ *in* $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$*, then* p*, the same sequence of variables, forms a possibly m-connecting path between A and B given* $\mathbf{C}$ *in* $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$.[15]

Proof See Appendix B. ■

If there is no possibly m-connecting path between A and B given $\mathbf{C}$ in a partial mixed graph, we say that A and B are *definitely m-separated* by $\mathbf{C}$ in the graph. A *do*-calculus follows:

Theorem 17 (*do*-calculus given a PAG) *Let* $\mathcal{P}$ *be the causal PAG for* $\mathbf{O}$*, and* $\mathbf{U}, \mathbf{X}, \mathbf{Y}, \mathbf{W}$ *be disjoint subsets of* $\mathbf{O}$*. The following rules are valid:*

1. *if* $\mathbf{Y}$ *and* $\mathbf{X}$ *are definitely m-separated by* $\mathbf{U} \cup \mathbf{W}$ *in* $\mathcal{P}_{\overline{\mathbf{U}}}$*, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w}).$$

2. *if* $\mathbf{Y}$ *and* $\mathbf{X}$ *are definitely m-separated by* $\mathbf{U} \cup \mathbf{W}$ *in* $\mathcal{P}_{\underline{\mathbf{X}}\overline{\mathbf{U}}}$*, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}).$$

3. *if* $\mathbf{Y}$ *and* $\mathbf{X}$ *are definitely m-separated by* $\mathbf{U} \cup \mathbf{W}$ *in* $\mathcal{P}_{\overline{\mathbf{U}\mathbf{X}'}}$*, then*

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{w})$$

where $\mathbf{X}' = \mathbf{X} \backslash \mathbf{PossibleAn}_{\mathcal{P}_{\overline{\mathbf{U}}}}(\mathbf{W})$.

Proof It follows from Lemma 16 and Theorem 14. The only caveat is that in general $\mathbf{An}_{\mathcal{M}_{\overline{\mathbf{U}}}}(\mathbf{W}) \neq \mathbf{PossibleAn}_{\mathcal{P}_{\overline{\mathbf{U}}}}(\mathbf{W})$ for an arbitrary $\mathcal{M}$ represented by $\mathcal{P}$. But it is always the case that $\mathbf{An}_{\mathcal{M}_{\overline{\mathbf{U}}}}(\mathbf{W}) \subseteq \mathbf{PossibleAn}_{\mathcal{P}_{\overline{\mathbf{U}}}}(\mathbf{W})$, which means that $\mathbf{X} \backslash \mathbf{An}_{\mathcal{M}_{\overline{\mathbf{U}}}}(\mathbf{W}) \supseteq \mathbf{X} \backslash \mathbf{PossibleAn}_{\mathcal{P}_{\overline{\mathbf{U}}}}(\mathbf{W})$ for every $\mathcal{M}$ represented by $\mathcal{P}$. So it is possible that for rule (3), $\mathcal{P}_{\overline{\mathbf{U}\mathbf{X}'}}$ leaves more edges in than necessary, but it does not affect the validity of rule (3). ■

The possibility that $\mathcal{P}_{\overline{\mathbf{U}\mathbf{X}'}}$ leaves more edges in than necessary is one of three aspects in which our *do*-calculus may be "incomplete" in the following sense: it is possible that a rule in the PAG-based *do*-calculus is not applicable, but for every DAG compatible with the given PAG, the corresponding rule in Pearl's DAG-based calculus is applicable. The other two aspects are already noted: (1) the calculus is formulated in terms of the absence of possibly m-connecting paths (cf. Footnote 14, and more on this in the next section); and (2) the MAG-based *do*-calculus is based on Corollary 13 whose converse does not hold. Therefore, the PAG-based *do*-calculus as currently formulated may be further improved.

That said, let us illustrate the utility of the *do*-calculus with the simple example used in Section 3. Given the PAG in Figure 4 we can infer that $P(L \mid do(S), G) = P(L \mid S, G)$ by rule 2, because L and S are definitely m-separated by $\{G\}$ in $\mathcal{P}_{\underline{S}}$ (Figure 9(a)); and

15. For our purpose, what we need is the obvious consequence of the lemma that if there is an m-connecting path in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, then there is a possibly m-connecting path in $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. We suspect that a stronger result might hold as well: if there is an m-connecting path in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, then there is a definite m-connecting path in $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. We can't prove or disprove the stronger result at the moment.

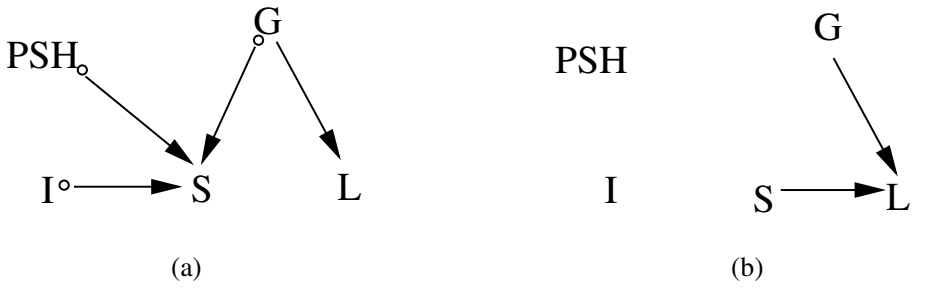


Figure 9: PAG Surgery: $\mathcal{P}_{\underline{S}}$ and $\mathcal{P}_{\overline{S}}$.

$P(G \mid do(S)) = P(G)$ by rule 3, because G and S are definitely m-separated in $\mathcal{P}_{\overline{S}}$ (Figure 9(b)). It follows that

$$\begin{aligned} P(L \mid do(S)) &= \sum_G P(L, G \mid do(S)) \\ &= \sum_G P(L \mid do(S), G) P(G \mid do(S)) \\ &= \sum_G P(L \mid S, G) P(G). \end{aligned}$$

By contrast, it is not valid in the *do*-calculus that $P(L \mid do(G), S) = P(L \mid G, S)$ because L and G are not definitely m-separated by $\{S\}$ in $\mathcal{P}_{\underline{G}}$, which is depicted in Figure 10. (Notice the bi-directed edge between L and G.)

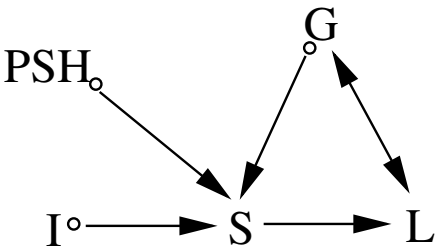


Figure 10: PAG Surgery: $\mathcal{P}_{\underline{G}}$.

5. Invariance Under Interventions

We now develop stronger results for a key component of *do*-calculus, the property of *invariance under interventions*, first systematically studied in Spirtes et al. (1993). The idea is simple. A conditional probability $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{Z} = \mathbf{z})$ is said to be *invariant* under an intervention $\mathbf{X} := \mathbf{x}$—or $do(\mathbf{X} = \mathbf{x})$—if $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{z}) = P(\mathbf{y} \mid \mathbf{z})$.[16] This concept (under

16. Here we allow that $\mathbf{X}$ and $\mathbf{Z}$ have a non-empty intersection, and assume that the conditioning operation is applied to the post-intervention population (i.e., intervening comes before conditioning). As a result,

the name of 'observability') plays an important role in some interesting theoretical work on observational studies (e.g., Pratt and Schlaifer, 1988; for a good review see Winship and Morgan, 1999), and also forms the basis of the prediction algorithm presented in Spirtes et al. (1993), which seeks to identify a post-intervention probability by searching for an expression in terms of invariant probabilities.

It is also the corner stone of Pearl's *do*-calculus. To see this, let us take a closer look at the second and third rules in the *do*-calculus. The second rule of the calculus gives a graphical condition for when we can conclude

$$P(\mathbf{y} \mid do(\mathbf{u}), do(\mathbf{x}), \mathbf{w}) = P(\mathbf{y} \mid do(\mathbf{u}), \mathbf{x}, \mathbf{w}).$$

If we take $\mathbf{U}$ to be the empty set and write the above equation in the subscript notation, we get

$$P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{w}) = P(\mathbf{y} \mid \mathbf{x}, \mathbf{w}).$$

Since $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{X} = \mathbf{x}) = 1$, thanks to the supposed effectiveness of the intervention, we have

$$P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{w}) = P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}).$$

So a special case of the second rule is a condition for $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = P(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$, that is, for when $P(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ is invariant under the intervention $\mathbf{X} := \mathbf{x}$. In fact, the second rule is nothing but a generalization of this condition to tell when a post-intervention probability $P_{\mathbf{u}}(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ would be invariant under a *further* intervention $\mathbf{X} := \mathbf{x}$.

The third rule is more obviously about invariance. It is a generalization of the condition for $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{w}) = P(\mathbf{y} \mid \mathbf{w})$, that is, for when $P(\mathbf{y} \mid \mathbf{w})$ is invariant under the intervention $\mathbf{X} := \mathbf{x}$. The difference between rule 2 and rule 3 is that rule 2 is about invariance of $P(\mathbf{y} \mid \mathbf{z})$ under an intervention on $\mathbf{X}$ in case $\mathbf{X} \subseteq \mathbf{Z}$ $(= \mathbf{X} \cup \mathbf{W})$, whereas rule 3 is about invariance of $P(\mathbf{y} \mid \mathbf{z})$ under an intervention on $\mathbf{X}$ in case $\mathbf{X}$ and $\mathbf{Z}$ $(= \mathbf{W})$ are disjoint. As we mentioned earlier, the first rule is not essential, so the *do*-calculus is in effect a generalization of conditions for invariance.

We now focus on this key component of *do*-calculus, and present better graphical conditions for judging invariance given a PAG than those that are implied by the PAG-based *do*-calculus presented in the last section. The conditions for invariance implied by Pearl's (DAG-based) *do*-calculus can be equivalently formulated without referring to manipulated graphs, as given in Spirtes et al. (1993, Theorem 7.1) before the *do*-calculus was invented. In this section we develop corresponding conditions in terms of PAGs. The conditions will be not only sufficient in the sense that if the conditions are satisfied, then every DAG compatible with the given PAG entails invariance, but also necessary in the sense that if the conditions fail, then there is at least one DAG compatible with the given PAG that does not entail invariance. In this aspect, the conditions are also superior to earlier results on invariance given an equivalence class of DAGs due to Spirtes et al. (1993, Theorems 7.3 and 7.4).

We first state the conditions for judging invariance given a DAG, originally presented in Spirtes et al. (1993, Theorem 7.1).

Proposition 18 (Spirtes, Glymour, Scheines) *Let $\mathcal{G}$ be the causal DAG for $\mathbf{O} \cup \mathbf{L}$, and $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$ be three sets of variables such that $\mathbf{X} \cap \mathbf{Y} = \mathbf{Y} \cap \mathbf{Z} = \emptyset$ (but $\mathbf{X}$ and $\mathbf{Z}$ can overlap). $P(\mathbf{y} \mid \mathbf{z})$ is invariant under an intervention on $\mathbf{X}$ if*

when we speak of $P_{\mathbf{X}:=\mathbf{x}}(\mathbf{y} \mid \mathbf{z})$, we implicitly assume that $\mathbf{x}$ and $\mathbf{z}$ are consistent regarding the values for variables in $\mathbf{X} \cap \mathbf{Z}$, for otherwise the quantity is undefined.

(1) *for every* $X \in \mathbf{X} \cap \mathbf{Z}$, *there is no d-connecting path between* X *and any member of* $\mathbf{Y}$ *given* $\mathbf{Z} \backslash \{X\}$ *that is into* X;

(2) *for every* $X \in \mathbf{X} \cap (\mathbf{An}_{\mathcal{G}}(\mathbf{Z}) \backslash \mathbf{Z})$, *there is no d-connecting path between* X *and any member of* $\mathbf{Y}$ *given* $\mathbf{Z}$; *and*

(3) *for every* $X \in \mathbf{X} \backslash \mathbf{An}_{\mathcal{G}}(\mathbf{Z})$, *there is no d-connecting path between* X *and any member of* $\mathbf{Y}$ *given* $\mathbf{Z}$ *that is out of* X.[17]

Remark: Because $\mathbf{Z} \subseteq \mathbf{An}_{\mathcal{G}}(\mathbf{Z})$, $\mathbf{X} \cap \mathbf{Z}$, $\mathbf{X} \cap (\mathbf{An}_{\mathcal{G}}(\mathbf{Z}) \backslash \mathbf{Z})$ and $\mathbf{X} \backslash \mathbf{An}_{\mathcal{G}}(\mathbf{Z})$ form a partition of $\mathbf{X}$. So for each member of $\mathbf{X}$, only one of the conditions is relevant.

The proposition is an equivalent formulation of Theorem 7.1 in Spirtes et al. (1993). It is not hard to check that the proposition follows from rules 2 and 3 in the DAG-based *do*-calculus (Proposition 7); the talk of d-separation in manipulated graphs is replaced by the talk of absence of d-connecting paths of certain orientations in the original graph. Conversely, the proposition implies the special case of rules 2 and 3 where the background intervention $do(\mathbf{U})$ is empty. Specifically, clause (1) in the proposition corresponds to rule 2 in the *do*-calculus; clauses (2) and (3) correspond to rule 3 in the *do*-calculus.

Spirtes et al. (1993, pp. 164–5) argued that these conditions are also "almost necessary" for invariance. What they meant is that if the conditions are not satisfied, then the causal structure does not *entail* the invariance, although there may exist some particular distribution compatible with the causal structure such that $P(\mathbf{y} \mid \mathbf{z})$ is invariant under some particular intervention on $\mathbf{X}$. From now on when we speak of invariance entailed by the causal DAG, we mean that the conditions in Proposition 18 are satisfied—or equivalently, that the invariance follows from an application of rule 2 or rule 3 in the DAG-based *do*-calculus.[18] Our purpose is to demonstrate that there are corresponding graphical conditions relative to a PAG that are sufficient and necessary for the conditions in Proposition 18 to hold for each and every DAG compatible with the PAG.

Once again, we develop the conditions in two steps: first to MAGs and then to PAGs. In the first step, our goal is to find sufficient and necessary conditions for invariance entailed by a MAG, as defined below:

Definition 19 (Invariance entailed by a MAG) *Let* $\mathcal{M}$ *be a causal MAG over* $\mathbf{O}$, *and* $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z} \subseteq \mathbf{O}$ *be three sets of variables such that* $\mathbf{X} \cap \mathbf{Y} = \mathbf{Y} \cap \mathbf{Z} = \emptyset$, $P(\mathbf{y} \mid \mathbf{z})$ *is* ***entailed to be***

17. It is not hard to see that (3) is equivalent to saying that for every $X \in \mathbf{X} \backslash \mathbf{An}_{\mathcal{G}}(\mathbf{Z})$, there is no directed path from X to any member of $\mathbf{Y}$. Lemma 23 below is an immediate corollary of this equivalent formulation.
18. This stipulation is of course not intended to be a definition of the notion of *structurally entailed invariance*. A proper definition would be to the effect that for every distribution compatible with the causal structure, $P(\mathbf{y} \mid \mathbf{z})$ is invariant under any intervention of $\mathbf{X}$. The argument given by Spirtes et al. (1993, pp. 164–5) for (their equivalent formulation of) Proposition 18 suggests that the conditions are sufficient and necessary for structurally entailed invariance. Their argument uses the device of what they call policy variables, extra variables introduced into the causal DAG to represent interventions. Given the causal DAG $\mathcal{G}$, a policy variable for a variable X is an (extra) parent of X but otherwise not adjacent to any other variables in $\mathcal{G}$. Interventions can then be simulated by conditioning on the intervention variables, and invariance can be reformulated as conditional independence involving intervention variables. The conditions in Proposition 18 are equivalent to saying that the variables in $\mathbf{Y}$ are d-separated from the policy variables for $\mathbf{X}$ by $\mathbf{Z}$ (in the graph augmented by the policy variables). It thus seems plausible that these conditions are sufficient and necessary for structurally entailed invariance, given that d-separation is a sufficient and necessary condition for structurally entailed conditional independence (Geiger et al., 1990; Meek, 1995b). But Spirtes et al. did not give a rigorous proof for necessity. As an anonymous reviewer points out, the rigorous proof, if any, would need to be carefully made, and in particular, one should be careful in treating policy variables as random variables. We will not take on this task here.

invariant under interventions on $\mathbf{X}$ ***given*** $\mathcal{M}$ *if for every DAG* $\mathcal{G}(\mathbf{O}, \mathbf{L})$ *represented by* $\mathcal{M}$, $P(\mathbf{y} \mid \mathbf{z})$ *is entailed to be invariant under interventions on* $\mathbf{X}$ *given* $\mathcal{G}$ *(i.e., the conditions in Proposition 18 are satisfied).*

The question is how to judge invariance entailed by a MAG without doing the intractable job of checking the conditions in Proposition 18 for each and every compatible DAG. The next few lemmas, Lemmas 20–23, state useful connections between d-connecting paths in a DAG and m-connecting paths in the corresponding MAG. Lemma 20 records the important result due to Richardson and Spirtes (2002) that d-separation relations among observed variables in a DAG with latent variables correspond exactly to m-separation relations in its MAG.

Lemma 20 *Let* $\mathcal{G}$ *be any DAG over* $\mathbf{O} \cup \mathbf{L}$, *and* $\mathcal{M}$ *be the MAG of* $\mathcal{G}$ *over* $\mathbf{O}$. *For any* $A, B \in \mathbf{O}$ *and* $\mathbf{C} \subseteq \mathbf{O}$ *that does not contain* A *or* B, *there is a path d-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{G}$ *if and only if there is a path m-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{M}$.

Proof This is a special case of Lemma 17 and Lemma 18 in Spirtes and Richardson (1996), and also a special case of Theorem 4.18 in Richardson and Spirtes (2002). ■

Given Lemma 20, we know how to tell whether clause (2) of Proposition 18 holds in all DAGs compatible with a given MAG. For the other two conditions in Proposition 18, we need to take into account the orientations of d-connecting paths.

Lemma 21 *Let* $\mathcal{G}$ *be any DAG over* $\mathbf{O} \cup \mathbf{L}$, *and* $\mathcal{M}$ *be the MAG of* $\mathcal{G}$ *over* $\mathbf{O}$. *For any* $A, B \in \mathbf{O}$ *and* $\mathbf{C} \subseteq \mathbf{O}$ *that does not contain* A *or* B, *if there is a path d-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{G}$ *that is into* A, *then there is a path m-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{M}$ *that is either into* A *or contains an invisible edge out of* A.

Proof See Appendix B. ■

Lemma 22 *Let* $\mathcal{M}$ *be any MAG over* $\mathbf{O}$. *For any* $A, B \in \mathbf{O}$ *and* $\mathbf{C} \subseteq \mathbf{O}$ *that does not contain* A *or* B, *if there is a path m-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{M}$ *that is either into* A *or contains an invisible edge out of* A, *then there exists a DAG* $\mathcal{G}$ *over* $\mathbf{O} \cup \mathbf{L}$ *(for some extra variables* $\mathbf{L}$*) whose MAG is* $\mathcal{M}$, *such that in* $\mathcal{G}$ *there is a path d-connecting* A *and* B *given* $\mathbf{C}$ *that is into* A.

Proof See Appendix B. ■

Obviously these two lemmas are related to adapting clause (1) in Proposition 18 to MAGs. The next lemma is related to clause (3).

Lemma 23 *Let* $\mathcal{G}$ *be any DAG over* $\mathbf{O} \cup \mathbf{L}$, *and* $\mathcal{M}$ *be the MAG of* $\mathcal{G}$ *over* $\mathbf{O}$. *For any* $A, B \in \mathbf{O}$ *and* $\mathbf{C} \subseteq \mathbf{O}$ *that does not contain* B *or any descendant of* A *in* $\mathcal{G}$ *(or in* $\mathcal{M}$, *since* $\mathcal{G}$ *and* $\mathcal{M}$ *have the same ancestral relations among variables in* $\mathbf{O}$*), there is a path d-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{G}$ *that is out of* A *if and only if there is a path m-connecting* A *and* B *given* $\mathbf{C}$ *in* $\mathcal{M}$ *that is out of* A.

Proof See Appendix B. ■

Given these lemmas, the conditions in Proposition 18 are readily translated into the following conditions for invariance given a MAG.

Theorem 24 *Suppose $\mathcal{M}$ is the causal MAG over a set of variables $\mathbf{O}$. For any $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$, $\mathbf{X} \cap \mathbf{Y} = \mathbf{Y} \cap \mathbf{Z} = \emptyset$, $P(\mathbf{y} \mid \mathbf{z})$ is entailed to be invariant under interventions on $\mathbf{X}$ given $\mathcal{M}$ if and only if*

(1) for every $X \in \mathbf{X} \cap \mathbf{Z}$, there is no m-connecting path between X and any member of $\mathbf{Y}$ given $\mathbf{Z} \backslash \{X\}$ that is into X or contains an invisible edge out of X;

(2) for every $X \in \mathbf{X} \cap (\mathbf{An}_{\mathcal{M}}(\mathbf{Z}) \backslash \mathbf{Z})$, there is no m-connecting path between X and any member of $\mathbf{Y}$ given $\mathbf{Z}$; and

(3) for every $X \in \mathbf{X} \backslash \mathbf{An}_{\mathcal{M}}(\mathbf{Z})$, there is no m-connecting path between X and any member of $\mathbf{Y}$ given $\mathbf{Z}$ that is out of X.

Proof Given Lemma 21, if (1) holds, then for every DAG represented by $\mathcal{M}$, the first condition in Proposition 18 holds. Given Lemma 20 and the fact that $\mathcal{M}$ and all DAGs represented by $\mathcal{M}$ have the exact same ancestral relations among $\mathbf{O}$, if (2) holds, the second condition in Proposition 18 holds for every DAG represented by $\mathcal{M}$. Moreover, given Lemma 23, if (3) holds, the third condition in Proposition 18 holds for every DAG represented by $\mathcal{M}$. So (1), (2) and (3) together imply that $P(\mathbf{y} \mid \mathbf{z})$ is invariant under interventions on $\mathbf{X}$ given $\mathcal{M}$.

Conversely, if (1) fails, then by Lemma 22, there is a DAG represented by $\mathcal{M}$ in which the first condition in Proposition 18 fails. Likewise with conditions (2) and (3), in light of Lemmas 20 and 23 and the fact that $\mathcal{M}$ and a DAG represented by $\mathcal{M}$ have the exact same ancestral relations among $\mathbf{O}$. So (1), (2) and (3) are also necessary for $P(\mathbf{y} \mid \mathbf{z})$ to be entailed to be invariant under interventions on $\mathbf{X}$ given $\mathcal{M}$. ■

For example, given the MAG in Figure 3(a), $P(L \mid G, S)$ is invariant under interventions on S, because the only m-connecting path between L and S given G is $\langle L, S \rangle$, which contains a visible directed edge out of L, and so the relevant clause in Theorem 24, clause (1), is satisfied. By contrast, $P(L \mid G, S)$ is not entailed to be invariant under interventions on G given the MAG—in the sense that there exists a causal DAG compatible with the MAG given which $P(L \mid G, S)$ is not entailed to be invariant under interventions on G—because clause (1) is not satisfied.

In a similar fashion, we can extend the result to invariance entailed by a PAG. Definition first:

Definition 25 (Invariance entailed by a PAG) *Let $\mathcal{P}$ be a PAG over $\mathbf{O}$, and $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$ be three sets of variables such that $\mathbf{X} \cap \mathbf{Y} = \mathbf{Y} \cap \mathbf{Z} = \emptyset$, $P(\mathbf{y} \mid \mathbf{z})$ is **entailed to be invariant under interventions on $\mathbf{X}$ given** $\mathcal{P}$ if for every MAG $\mathcal{M}$ in the Markov equivalence class represented by $\mathcal{P}$, $P(\mathbf{y} \mid \mathbf{z})$ is entailed to be invariant under interventions on $\mathbf{X}$ given $\mathcal{M}$.*

We need a few lemmas that state connections between m-connecting paths in a MAG and definite m-connecting paths (as opposed to mere possibly m-connecting paths) in its PAG. By the definition of definite m-connecting paths (Definition 4), definite m-connection in a PAG implies m-connection in every MAG represented by the PAG. It is not obvious, however, that m-connection in a MAG will always be revealed as definite m-connection in its PAG. Fortunately, this turns out to be true. However, the proof is highly involved, and relies on many results about the properties of PAGs and the transformation between PAGs and MAGs presented in Zhang (2006, chapters 3–4), which would take up too much space and might distract readers from the main points of the present paper. So we will simply state the fact here, and refer interested readers to Zhang (2006, chap. 5, Lemma 5.1.7) for the proof.

Lemma 26 *Let $\mathcal{M}$ be a MAG over $\mathbf{O}$, and $\mathcal{P}$ be the PAG that represents $[\mathcal{M}]$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain A or B, if there is a path m-connecting A and B given $\mathbf{C}$ in $\mathcal{M}$, then there is a path definitely m-connecting A and B given $\mathbf{C}$ in $\mathcal{P}$. Furthermore, if there is an m-connecting path in $\mathcal{M}$ that is either into A or out of A with an invisible directed edge, then there is a definite m-connecting path in $\mathcal{P}$ that does not start with a definitely visible edge out of A.*

Proof See the proof of Lemma 5.1.7 in Zhang (2006, pp. 207). ■

The converse to the second part of Lemma 26 is also true.

Lemma 27 *Let $\mathcal{P}$ be a PAG over $\mathbf{O}$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain A or B, if there is a path definitely m-connecting A and B given $\mathbf{C}$ in $\mathcal{P}$ that does not start with a definitely visible edge out of A, then there exists a MAG $\mathcal{M}$ in the equivalence class represented by $\mathcal{P}$ in which there is a path m-connecting A and B given $\mathbf{C}$ that is either into A or includes an invisible directed edge out of A.*

Proof See Appendix B. ■

Lemmas 26 and 27 are useful for establishing conditions analogous to clauses (1) and (2) in Theorem 24. For clause (3), we need two more lemmas.

Lemma 28 *Let $\mathcal{M}$ be a MAG over $\mathbf{O}$, and $\mathcal{P}$ be the PAG that represents $[\mathcal{M}]$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain B or any descendant of A in $\mathcal{M}$, if there is a path m-connecting A and B given $\mathbf{C}$ in $\mathcal{M}$ that is out of A, then there is a path definitely m-connecting A and B given $\mathbf{C}$ in $\mathcal{P}$ that is not into A (i.e., the edge incident to A on the path is either $A \circ\!\!-\!\!\circ$, or $A \circ\!\!\rightarrow$, or $A \rightarrow$).*

Proof See Appendix B. ■

Lemma 29 *Let $\mathcal{P}$ be a PAG over $\mathbf{O}$. For any $A, B \in \mathbf{O}$ and $\mathbf{C} \subseteq \mathbf{O}$ that does not contain A or B, if there is a path definitely m-connecting A and B given $\mathbf{C}$ in $\mathcal{P}$ that is not into A, then there exists a MAG $\mathcal{M}$ represented by $\mathcal{P}$ in which there is a path m-connecting A and B given $\mathbf{C}$ that is out of A.*

Proof See Appendix B. ■

The main theorem follows.

Theorem 30 *Suppose $\mathcal{P}$ is the causal PAG over a set of variables $\mathbf{O}$. For any $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}$ such that $\mathbf{X} \cap \mathbf{Y} = \mathbf{Y} \cap \mathbf{Z} = \emptyset$, $P(\mathbf{y} \mid \mathbf{z})$ is entailed to be invariant under interventions on $\mathbf{X}$ given $\mathcal{P}$ if and only if*

(1) for every $X \in \mathbf{X} \cap \mathbf{Z}$, every definite m-connecting path, if any, between X and any member of $\mathbf{Y}$ given $\mathbf{Z} \backslash \{X\}$ is out of X with a definitely visible edge;

(2) for every $X \in \mathbf{X} \cap (\mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z}) \backslash \mathbf{Z})$, there is no definite m-connecting path between X and any member of $\mathbf{Y}$ given $\mathbf{Z}$; and

(3) for every $X \in \mathbf{X} \backslash \mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z})$, every definite m-connecting path, if any, between X and any member of $\mathbf{Y}$ given $\mathbf{Z}$ is into X.

Proof We show that (1), (2) and (3) are sufficient and necessary for the corresponding conditions in Theorem 24 to hold for all MAGs represented by $\mathcal{P}$. It follows from Lemma 26 that if (1) holds, then the first condition in Theorem 24 holds for all MAGs represented by $\mathcal{P}$. Note moreover that for every MAG $\mathcal{M}$ represented by $\mathcal{P}$, $\mathbf{An}_{\mathcal{M}}(\mathbf{Z}) \subseteq \mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z})$. It again follows from Lemma 26 that if (2) holds, then the second condition in Theorem 24 holds for all MAGs represented by $\mathcal{P}$. Finally, it follows from Lemma 28 (and Lemma 26) that if (3) holds, the third condition in Theorem 24 holds for all MAGs represented by $\mathcal{P}$. Hence (1), (2) and (3) are sufficient.

Conversely, if (1) fails, then by Lemma 27, there exists a MAG represented by $\mathcal{P}$ for which the first condition in Theorem 24 fails.

To show the necessity of (2), we need the fact mentioned in Footnote 11 that if X is a possible ancestor of a vertex $Z \in \mathbf{Z}$ in $\mathcal{P}$, then there exists a MAG represented by $\mathcal{P}$, in which X is an ancestor of Z. So if (2) fails, that is, there is a definite m-connecting path between a variable $X \in \mathbf{X} \cap (\mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z}) \backslash \mathbf{Z})$ and a member of $\mathbf{Y}$ given $\mathbf{Z}$ in $\mathcal{P}$, then there exists a MAG $\mathcal{M}$ represented by $\mathcal{P}$ in which $X \in \mathbf{X} \cap (\mathbf{An}_{\mathcal{M}}(\mathbf{Z}) \backslash \mathbf{Z})$, and there is an m-connecting path between X and a member of $\mathbf{Y}$ given $\mathbf{Z}$, which violates clause (2) of Theorem 24.

Lastly, if (3) fails, that is, there is a definite m-connecting path between a variable $X \in \mathbf{X} \backslash \mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z})$ and a member of $\mathbf{Y}$ given $\mathbf{Z}$ that is *not* into X, then it follows from Lemma 29 that there exists a MAG $\mathcal{M}$ represented by $\mathcal{P}$ in which there is an m-connecting path between X and a member of $\mathbf{Y}$ given $\mathbf{Z}$ that is out of X. Moreover, since $X \in \mathbf{X} \backslash \mathbf{PossibleAn}_{\mathcal{P}}(\mathbf{Z})$, X cannot be an ancestor of $\mathbf{Z}$ in $\mathcal{M}$, that is, $X \in \mathbf{X} \backslash \mathbf{An}_{\mathcal{M}}(\mathbf{Z})$. So $\mathcal{M}$ fails clause (3) of Theorem 24. Therefore, the conditions are also necessary. ■

For a simple illustration, consider again the PAG in Figure 4. Given the PAG, it can be inferred that $P(L \mid G, S)$ is invariant under interventions on I, because there is no definite m-connecting path between L and I given $\{G, S\}$, satisfying the relevant clause—clause (2)—in Theorem 30. $P(L \mid G, S)$ is also invariant under interventions on S because the only definitely m-connecting path between L and S given $\{G\}$ is $S \rightarrow L$ which contains a definitely visible edge out of S, satisfying the relevant clause—clause (1)—in Theorem 30.

On the other hand, for example, $P(S)$ is not entailed to be invariant under interventions on I. Note that given the MAG of Figure 3(b), $P(S)$ is indeed entailed to be invariant under interventions on I, but this invariance is not unanimously implied in the equivalence class. Given some other MAGs in the class, such as the one in Figure 3(a), $P(S)$ is not entailed to be invariant under interventions on I.

As briefly noted in the last section, the PAG-based *do*-calculus in its current form is not complete. We mentioned three issues that might be responsible for this (cf. the comments right after Theorem 17), but only one of them we are sure leads to counterexamples—examples in which a rule in the DAG-based calculus is applicable for all DAGs compatible with the given PAG, but the corresponding rule in the PAG-based calculus is not applicable. It is the fact that the calculus is formulated in terms of absence of possibly m-connecting paths. Consider the example we used to illustrate the difference between definite and possibly m-connecting paths in Section 3. Given the PAG in Figure 5, we cannot apply rule 2 of the PAG-based *do*-calculus to conclude that $P(W \mid do(X), Y, Z) = P(W \mid Y, Z)$, because there is a possibly m-connecting path between X and W relative to $\{Y, Z\}$ in the PAG (note that since $X \in \mathbf{PossibleAn}(\{Y, Z\})$, the rule does not require manipulating the graph). However, it can be shown that for every DAG compatible with the PAG, X and W are d-separated by $\{Y, Z\}$ in either the

X-upper-manipulation of the DAG or in the DAG itself. So rule 2 of the DAG-based *do*-calculus is actually applicable given any DAG compatible with the PAG.

Although we suspect that such counterexamples may not be encountered often in practice, it is at least theoretically interesting to handle them. Our results in this section provide an improvement in regard to the important special case of invariance. That is, the conditions given in Theorem 30 are complete for deriving statements of invariance, in the following sense: if the conditions therein fail relative to a PAG, then there exists a DAG represented by the PAG given which the conditions in Proposition 18 do not hold. The example in Figure 5 is not a counterexample to the completeness of Theorem 30. Unlike the *do*-calculus presented in Theorem 17, Theorem 30 implies that $P(W \mid Y, Z)$ is entailed to be invariant under interventions on X given the PAG (and hence we can conclude that $P(W \mid do(X), Y, Z) = P(W \mid Y, Z)$), because there is no definite m-connecting path between X and W relative to $\{Y, Z\}$ in the PAG. Whether it is valid to formulate the PAG-based *do*-calculus in terms of definite m-connecting paths is an open question at this point (cf. Footnote 15).[19]

Theorem 30 is in style very similar to Theorems 7.3 and 7.4 in Spirtes et al. (1993). The latter are formulated with respect to a *partially oriented inducing path graph* (POIPG). We include in Appendix A a description of the inducing path graphs (IPGs) as well as their relationship to ancestral graphs. As shown there, syntactically the class of ancestral graphs is a proper subclass of the class of inducing path graphs. In consequence a PAG in general reveals more qualitative causal information than a POIPG. In addition, it seems MAGs are easier to parameterize than IPGs. (For a linear parametrization of MAGs, see Richardson and Spirtes (2002).)

Apart from the advantages of working with MAGs and PAGs over IPGs and POIPGs, our Theorem 30 is superior to Spirtes et al.'s theorems in that our theorem is formulated in terms of definite m-connecting paths, whereas theirs, like the results in the last section, are formulated in terms of possibly m-connecting paths. As a result, their conditions are only sufficient but not necessary. Regarding the case in Figure 5, for example, their theorems do not imply that $P(W \mid Y, Z)$ is entailed to be invariant under interventions on X, due to the presence of the possibly m-connecting path in the graph (which in this case is also the POIPG). Furthermore, since definite m-connecting paths are special cases of possibly m-connecting paths, there are more possibly m-connecting paths than definite m-connecting paths to check in a PAG. This may turn out to be a computational advantage for our theorem.

6. Conclusion

Causal reasoning about consequences of interventions has received rigorous and interesting treatments in the framework of causal Bayesian networks. Much of the work assumes that the structure of the causal Bayesian network, represented by a directed acyclic graph, is fully given. In this paper we have provided some results about causal reasoning under weaker causal assumptions, represented by a maximal ancestral graph or a partial ancestral graph, the latter of which is fully testable with observational data (assuming the causal Faithfulness condition).

19. Here is another way to view the open problem. As explained earlier, *do*-calculus is essentially a generalization of the invariance conditions. Not only does it address the question of when $(\mathbf{y} \mid \mathbf{z})$ is invariant under an intervention $\mathbf{X} := \mathbf{x}$, it also addresses the more general question of when a post-intervention probability $P_{\mathbf{u}}(\mathbf{y} \mid \mathbf{z})$ would be invariant under a *further* intervention $\mathbf{X} := \mathbf{x}$. Our results in this section do not cover the latter question. To generalize the results in terms of definite m-connecting paths to address the latter question is parallel to improving the *do*-calculus.

Theorem 17 in Section 4 gives us a *do*-calculus under testable causal assumptions, represented by a PAG. The idea is that when any rule in the calculus is applicable given the PAG, the corresponding rule in Pearl's original *do*-calculus is applicable relative to each and every DAG compatible with the PAG. The converse, however, is not true; it is not the case that whenever all DAGs compatible with the PAG sanction the application of a certain rule in the *do*-calculus, the corresponding rule in the PAG-based calculus is also applicable. An interesting project is to either improve the calculus, or to investigate more closely the extent to which the current version is not complete.

As a first step towards improvement, we examined in Section 5 an important special case of the *do*-calculus—the graphical conditions for invariance under interventions—and presented sufficient and necessary conditions for invariance given a PAG. These conditions are very similar but also superior to the analogous results proved by Spirtes et al. (1993). In the latter work, there is also an algorithm (named Prediction Algorithm) for identifying post-intervention probabilities based on the conditions for invariance. The results in this paper can certainly be used to improve that algorithm.

The search for a syntactic derivation in the *do*-calculus to identify a post-intervention probability is no minor computational task. For this reason, it is worth deriving handy graphical criteria for identifiability from the *do*-calculus. Since invariant quantities are the most basic identifiable quantities, the condition for invariance is the most basic among such graphical criteria. Other graphical criteria in the literature, including the well known "back door criterion" and "front door criterion", should be extendible to PAGs in the same way as we did for invariance. On the other hand, a novel approach to identification has been developed recently by Tian and Pearl (2004), which proves computationally attractive. To adapt that approach to ancestral graphs is probably a worthy project.

Acknowledgments

I am grateful to Clark Glymour, Thomas Richardson, and Peter Spirtes for their helpful comments on the part of my dissertation this paper is based on. Thanks also to three anonymous referees for helping improve the paper significantly. One of them, especially, made extremely detailed and helpful suggestions.

Appendix A. Inducing Path Graphs

The theory of invariance under interventions developed in this paper is largely parallel to that developed in Spirtes et al. (1993). Their theory is based on a graphical representation called inducing path graphs. This graphical object is not given an independent syntactic definition, but defined via a construction relative to a DAG (with latent variables). It is clear from the construction that this representation is closely related to MAGs. In this appendix we specify the exact relationship between them. In particular, we justify an independent syntactic definition of inducing path graphs, which makes it clear that syntactically the class of MAGs is a subclass of inducing path graphs.

An *inducing path graph (IPG)* is a directed mixed graph, defined relative to a DAG, through the following construction:

Input: a DAG $\mathcal{G}$ over $\langle \mathbf{O}, \mathbf{L} \rangle$
Output: an IPG $\mathcal{I}_\mathcal{G}$ over $\mathbf{O}$

1. for each pair of variables $A, B \in \mathbf{O}$, A and B are adjacent in $\mathcal{I}_\mathcal{G}$ if and only if there is an inducing path between them relative to $\mathbf{L}$ in $\mathcal{G}$;

2. for each pair of adjacent vertices A, B in $\mathcal{I}_\mathcal{G}$, mark the A-end of the edge as an arrowhead if there is an inducing path between A and B that is into A, otherwise mark the A-end of the edge as a tail.

It can be shown that the construction outputs a mixed graph $\mathcal{I}_\mathcal{G}$ in which the set of m-separation relations matches exactly the set of d-separation relations among $\mathbf{O}$ in the original DAG $\mathcal{G}$ (Spirtes and Verma, 1992). Furthermore, $\mathcal{I}_\mathcal{G}$ encodes information about inducing paths in the original graph, which in turn implies features of the original DAG that bear causal significance. Specifically, we have two useful facts: (i) if there is an inducing path between A and B relative to $\mathbf{L}$ that is out of A, then A is an ancestor of B in $\mathcal{G}$; (ii) if there is an inducing path between A and B relative to $\mathbf{L}$ that is into both A and B, then A and B have a common ancestor in $\mathbf{L}$ unmediated by any other observed variable.[20] So $\mathcal{I}_\mathcal{G}$, just like the MAG for $\mathcal{G}$, represents both the conditional independence relations and (features of) the causal structure among the observed variables $\mathbf{O}$. Since the above construction produces a unique graph given a DAG $\mathcal{G}$, it is fair to call $\mathcal{I}_\mathcal{G}$ the IPG for $\mathcal{G}$.

Therefore a directed mixed graph over a set of variables is an IPG if it is the IPG for some DAG. We now show that a directed mixed graph is an IPG if and only if it is maximal and does not contain a directed cycle.

Theorem 31 *For any directed mixed graph $\mathcal{I}$ over a set of variables $\mathbf{O}$, there exists a DAG $\mathcal{G}$ over $\mathbf{O}$ and possibly some extra variables $\mathbf{L}$ such that $\mathcal{I} = \mathcal{I}_\mathcal{G}$—that is, $\mathcal{I}$ is the IPG for $\mathcal{G}$—if and only if*

(i1) There is no directed cycle in $\mathcal{I}$; and

(i2) $\mathcal{I}$ is maximal (i.e., there is no inducing path between two non-adjacent variables).

Proof We first show that the conditions are necessary (**only if**). Suppose there exists a DAG $\mathcal{G}(\mathbf{O}, \mathbf{L})$ whose IPG is $\mathcal{I}$. In other words, $\mathcal{I}$ is the output of the IPG construction procedure given $\mathcal{G}$. If there is any directed cycle in $\mathcal{I}$, say $c = \langle O_1, \ldots, O_n, O_1 \rangle$, then between any pair of adjacent nodes in the cycle, O_i and O_{i+1} ($1 \leq i \leq n$ and $O_{n+1} = O_1$), there is an inducing path between them in $\mathcal{G}$ relative to $\mathbf{L}$, which, by one of the facts mentioned earlier, implies that O_i is an ancestor of O_{i+1} in $\mathcal{G}$. Thus there would be a directed cycle in $\mathcal{G}$ as well, a contradiction. Therefore there is no directed cycle in $\mathcal{I}$. To show that it is also maximal, consider any two non-adjacent nodes A and B in $\mathcal{I}$. We show that there is no inducing path in $\mathcal{I}$ between A and B. Otherwise let $p = \langle A, O_1, \ldots, O_n, B \rangle$ be an inducing path. By the construction, there is an inducing path relative to $\mathbf{L}$ in $\mathcal{G}$ between A and O_1 that is into O_1, and an inducing path relative to $\mathbf{L}$ in $\mathcal{G}$ between B and O_n that is into O_n, and for every $1 \leq i \leq i-1$, there is an inducing path relative to $\mathbf{L}$ in $\mathcal{G}$ between O_i and O_{i+1} that is into both. By Lemma 32 in Appendix B, it follows that there is an inducing path between A and B relative to $\mathbf{L}$ in $\mathcal{G}$, and so A and B should be adjacent in $\mathcal{I}$, a contradiction. Therefore $\mathcal{I}$ is also maximal.

Next we demonstrate sufficiency (**if**). If the two conditions hold, construct a DAG $\mathcal{G}$ as follows: retain all the directed edges in $\mathcal{I}$, and for each bi-directed edge $A \leftrightarrow B$ in $\mathcal{I}$, introduce a latent variable L_{AB} in $\mathcal{G}$ and replace $A \leftrightarrow B$ with $A \leftarrow L_{AB} \rightarrow B$.[21] It is

20. For more details of the causal interpretation of IPGs, see Spirtes et al. (1993, pp. 130–138).
21. This is named *canonical DAG* in Richardson and Spirtes (2002).

easy to see that the resulting graph $\mathcal{G}$ is a DAG, as in $\mathcal{I}$ there is no directed cycle. We show that $\mathcal{I} = \mathcal{I}_{\mathcal{G}}$, the IPG for $\mathcal{G}$. For any pair of variables A and B in $\mathcal{I}$, there are four cases to consider:

Case 1: $A \rightarrow B$ is in $\mathcal{I}$. Then $A \rightarrow B$ is also in $\mathcal{G}$, so A and B are adjacent in $\mathcal{I}_{\mathcal{G}}$. In $\mathcal{I}_{\mathcal{G}}$, the edge between A and B is not $A \leftarrow B$, because otherwise B would have to be an ancestor of A in $\mathcal{G}$, a contradiction. The edge is not $A \leftrightarrow B$ either, because otherwise there would have to be a latent variable that is a parent of both A and B, which by the construction of $\mathcal{G}$ is not the case. So $A \rightarrow B$ is also in $\mathcal{I}_{\mathcal{G}}$.

Case 2: $A \leftarrow B$ is in $\mathcal{I}$. By the same argument as in *Case 1*, $A \leftarrow B$ is also in $\mathcal{I}_{\mathcal{G}}$.

Case 3: $A \leftrightarrow B$ is in $\mathcal{I}$. Then there is a L_{AB} such that $A \leftarrow L_{AB} \rightarrow B$ is in $\mathcal{G}$. Then obviously $\langle A, L_{AB}, B \rangle$ is an inducing path relative to $\mathbf{L}$ in $\mathcal{G}$ that is into both A and B, and hence $A \leftrightarrow B$ is also in $\mathcal{I}_{\mathcal{G}}$.

Case 4: A and B are not adjacent in $\mathcal{I}$. We show that they are not adjacent in $\mathcal{I}_{\mathcal{G}}$ either. For this, we only need to show that there is no inducing path between A and B relative to $\mathbf{L}$ in $\mathcal{G}$. Suppose otherwise that there is such an inducing path p between A and B in $\mathcal{G}$. Let $\langle A, O_1, \ldots, O_n, B \rangle$ be the sub-sequence of p consisting of all observed variables on p. By the definition of inducing path, all O_i's ($1 \leq i \leq n$) are colliders on p and are ancestors of either A or B. By the construction of $\mathcal{G}$, it is easy to see that O_i's are also ancestors of either A or B in $\mathcal{I}$. It is also easy to see that either $A \rightarrow O_1$ or $A \leftarrow L_{AO_1} \rightarrow O_1$ appears in $\mathcal{G}$, which implies that there is an edge between A and O_1 that is into O_1 in $\mathcal{I}$. Likewise, there is an edge between O_n and B that is into O_n in $\mathcal{I}$, and there is an edge between O_i and O_{i+1} that is into both in $\mathcal{I}$ for all $1 \leq i \leq n-1$. So $\langle A, O_1, \ldots, O_n, B \rangle$ constitutes an inducing path between A and B in $\mathcal{I}$, which contradicts the assumption that $\mathcal{I}$ is maximal. So there is no inducing path between A and B relative to $\mathbf{L}$ in $\mathcal{G}$, which means that A and B are not adjacent in $\mathcal{I}_{\mathcal{G}}$.

Therefore $\mathcal{I} = \mathcal{I}_{\mathcal{G}}$, the IPG for $\mathcal{G}$. ■

Given this theorem, it is clear that we can define IPGs in terms of $(i1)$ and $(i2)$. So a MAG is also an IPG, but an IPG is not necessarily a MAG, as the former may contain an almost directed cycle. The simplest IPG which is not a MAG is shown in Figure 11.

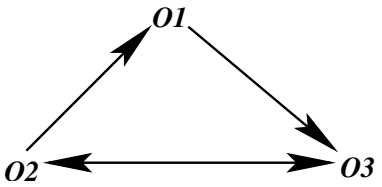


Figure 11: A simplest IPG that is not a MAG

Spirtes et al. (1993) uses *partially oriented inducing path graphs (POIPGs)* to represent Markov equivalence classes of IPGs. The idea is exactly the same as PAGs. A (complete) POIPG displays (all) common marks in a Markov equivalence class of IPGs. An obvious fact is that given a set of conditional independence facts that admits a faithful representation by a MAG, the Markov equivalence class of MAGs is included in the Markov equivalence class of IPGs. It follows that the POIPG cannot contain more informative marks than the PAG, but may contain fewer. So a PAG usually reveals more qualitative causal information than a POIPG does.

Appendix B. Proofs of the Lemmas

In proving some of the lemmas, we will use the following fact, which was proved in, for example, Spirtes et al. (1999, pp. 243):

Lemma 32 *Let $\mathcal{G}(\mathbf{O}, \mathbf{L})$ be a DAG, and $\langle V_0, \ldots, V_n \rangle$ be a sequence of distinct variables in $\mathbf{O}$. If (1) for all $0 \leq i \leq n-1$, there is an inducing path in $\mathcal{G}$ between V_i and V_{i+1} relative to $\mathbf{L}$ that is into V_i unless possibly $i = 0$ and is into V_{i+1} unless possibly $i = n-1$; and (2) for all $1 \leq i \leq n-1$, V_i is an ancestor of either V_0 or V_n in $\mathcal{G}$; then there is a subpath s of the concatenation of those inducing paths that is an inducing path between V_0 and V_n relative to $\mathbf{L}$ in $\mathcal{G}$. Furthermore, if the said inducing path between V_0 and V_1 is into V_0, then s is into V_0, and if the said inducing path between V_{n-1} and V_n is into V_n, then s is into V_n.*

Proof This is a special case of Lemma 10 in Spirtes et al. (1999, pp. 243). See their paper for a detailed proof. (One may think that the concatenation itself would be an inducing path between V_0 and V_n. This is almost correct, except that the concatenation may contain the same vertex multiple times. So in general it is a subsequence of the concatenation that constitutes an inducing path between V_0 and V_n.) ■

Lemma 32 gives a way to argue for the presence of an inducing path between two variables in a DAG, and hence is very useful for demonstrating that two variables are adjacent in the corresponding MAG. We will see several applications of this lemma in the subsequent proofs.

Proof of Lemma 9

Proof Since there is an inducing path between A and B relative to $\mathbf{L}$ in $\mathcal{G}$, A and B are adjacent in $\mathcal{M}$. Furthermore, since $A \in \mathbf{An}_{\mathcal{G}}(B)$, the edge between A and B in $\mathcal{M}$ is $A \to B$. We now show that it is invisible in $\mathcal{M}$. To show this, it suffices to show that for any C, if in $\mathcal{M}$ there is an edge between C and A that is into A or there is a collider path between C and A that is into A and every vertex on the path is a parent of B, then C is adjacent to B, which means that the condition for visibility cannot be met.

Let u be an inducing path between A and B relative to $\mathbf{L}$ in $\mathcal{G}$ that is into A. For any C, we consider the two possible cases separately:

Case 1: There is an edge between C and A in $\mathcal{M}$ that is into A. Then, by the way $\mathcal{M}$ is constructed from $\mathcal{G}$, there must be an inducing path u' in $\mathcal{G}$ between A and C relative to $\mathbf{L}$. Moreover, u' is into A, for otherwise A would be an ancestor of C, so that the edge between A and C in $\mathcal{M}$ would be out of A. Given u, u' and the fact that $A \in \mathbf{An}_{\mathcal{G}}(B)$, we can apply Lemma 32 to conclude that there is an inducing path between C and B relative to $\mathbf{L}$ in $\mathcal{G}$, which means C and B are adjacent in $\mathcal{M}$.

Case 2: There is a collider path p in $\mathcal{M}$ between C and A that is into A and every non-endpoint vertex on the path is a parent of B. For every pair of adjacent vertices $\langle V_i, V_{i+1} \rangle$ on p, the edge is $V_i \leftrightarrow V_{i+1}$ if $V_i \neq C$, and otherwise either $C \leftrightarrow V_{i+1}$ or $C \to V_{i+1}$. It follows that there is an inducing path in $\mathcal{G}$ between V_i and V_{i+1} relative to $\mathbf{L}$ such that the path is into V_{i+1}, and is into V_i unless possibly $V_i = C$. Given these inducing paths and the fact that every variable other than C on p is an ancestor of B, we can apply Lemma 32 to conclude that there is an inducing path between C and B relative to $\mathbf{L}$ in $\mathcal{G}$, which means C and B are adjacent in $\mathcal{M}$.

Therefore, the edge $A \to B$ is invisible in $\mathcal{M}$. ■

Proof of Lemma 10

Proof Construct a DAG from $\mathcal{M}$ as follows:

1. Leave every directed edge in $\mathcal{M}$ as it is. Introduce a latent variable L_{AB} and add $A \leftarrow L_{AB} \rightarrow B$ to the graph.

2. for every bi-directed edge $Z \leftrightarrow W$ in $\mathcal{M}$, delete the bi-directed edge. Introduce a latent variable L_{ZW} and add $Z \leftarrow L_{ZW} \rightarrow W$ to the graph.

The resulting graph we denote by $\mathcal{G}$, a DAG over $(\mathbf{O}, \mathbf{L})$, where $\mathbf{L} = \{L_{AB}\} \cup \{L_{ZW} \mid Z \leftrightarrow W \text{ is in } \mathcal{M}\}$. Obviously $\mathcal{G}$ is a DAG in which A and B share a latent parent. We need to show that $\mathcal{M} = \mathcal{M}_{\mathcal{G}}$, that is, $\mathcal{M}$ is the MAG of $\mathcal{G}$. For any pair of variables X and Y, there are four cases to consider:

Case 1: $X \rightarrow Y$ is in $\mathcal{M}$. Since $\mathcal{G}$ retains all directed edges in $\mathcal{M}$, $X \rightarrow Y$ is also in $\mathcal{G}$, and hence is also in $\mathcal{M}_{\mathcal{G}}$.

Case 2: $X \leftarrow Y$ is in $\mathcal{M}$. Same as *Case 1*.

Case 3: $X \leftrightarrow Y$ is in $\mathcal{M}$. Then there is a latent variable L_{XY} in $\mathcal{G}$ such that $X \leftarrow L_{XY} \rightarrow Y$ appears in $\mathcal{G}$. Since $X \leftarrow L_{XY} \rightarrow Y$ is an inducing path between X and Y relative to $\mathbf{L}$ in $\mathcal{G}$, X and Y are adjacent in $\mathcal{M}_{\mathcal{G}}$. Furthermore, it is easy to see that the construction of $\mathcal{G}$ does not create any directed path from X to Y or from Y to X. So X is not an ancestor of Y and Y is not an ancestor of X in $\mathcal{G}$. It follows that in $\mathcal{M}_{\mathcal{G}}$ the edge between X and Y is $X \leftrightarrow Y$.

Case 4: X and Y are not adjacent in $\mathcal{M}$. We show that in $\mathcal{G}$ there is no inducing path between X and Y relative to $\mathbf{L}$. Suppose otherwise that there is one. Let p be an inducing path between X and Y relative to $\mathbf{L}$ in $\mathcal{G}$ that includes a minimal number of observed variables. Let $\langle X, O_1, \ldots, O_n, Y \rangle$ be the sub-sequence of p consisting of all observed variables on p. By the definition of inducing path, all O_i's ($1 \leq i \leq n$) are colliders on p and are ancestors of either X or Y in $\mathcal{G}$. Since the construction of $\mathcal{G}$ does not create any new directed path from an observed variable to another observed variable, O_i's are also ancestors of either X or Y in $\mathcal{M}$. Since O_1 is a collider on p, either $X \rightarrow O_1$ or $X \leftarrow L_{XO_1} \rightarrow O_1$ appears in $\mathcal{G}$. Either way there is an edge between X and O_1 that is into O_1 in $\mathcal{M}$. Likewise, there is an edge between O_n and Y that is into O_n in $\mathcal{M}$.

Moreover, for all $1 \leq i \leq n-1$, the path p in $\mathcal{G}$ contains $O_i \leftarrow L_{O_iO_{i+1}} \rightarrow O_{i+1}$, because all O_i's are colliders on p. Thus in $\mathcal{M}$ there is an edge between O_i and O_{i+1}. Regarding these edges, by construction of the MAG, either all of them are bi-directed, or one of them is $A \rightarrow B$ and others are bi-directed. In the former case, $\langle X, O_1, \ldots, O_n, Y \rangle$ constitutes an inducing path between X and Y in $\mathcal{M}$, which contradicts the maximality of $\mathcal{M}$. In the latter case, without loss of generality, suppose $\langle A, B \rangle = \langle O_k, O_{k+1} \rangle$. Then $\langle X, O_1, ..., O_k = A \rangle$ is a collider path into A in $\mathcal{M}$. We now show by induction that for all $1 \leq i \leq k-1$, O_i is a parent of B in $\mathcal{M}$.

Consider O_{k-1} in the base case. O_{k-1} is adjacent to B, for otherwise $A \rightarrow B$ would be visible in $\mathcal{M}$ because there is an edge between O_{k-1} and A that is into A. The edge between O_{k-1} and B is not $O_{k-1} \leftarrow B$, for otherwise there would be $A \rightarrow B \rightarrow O_{k-1}$ and yet an edge between O_{k-1} and A that is into A in $\mathcal{M}$, which contradicts the fact that $\mathcal{M}$ is ancestral. The edge between them is not $O_{k-1} \leftrightarrow B$, for otherwise there would be an inducing path between X and Y relative to $\mathbf{L}$ in $\mathcal{G}$ that includes fewer observed variables than p does, which contradicts our choice of p. So O_{k-1} is a parent of B in $\mathcal{M}$.

In the inductive step, suppose for all $1 < m+1 \leq j \leq k-1$, O_j is a parent of B in $\mathcal{M}$, and we need to show that O_m is also a parent of B in $\mathcal{M}$. The argument is essentially the same as in the base case. Specifically, O_m and B are adjacent because otherwise it follows from the inductive hypothesis that $A \rightarrow B$ is visible. The edge is not $O_m \leftarrow B$ on pain of making $\mathcal{M}$ non-ancestral; and the edge is not $O_m \leftrightarrow B$ on pain of creating

an inducing path that includes fewer observed variables than p does. So O_m is also a parent of B.

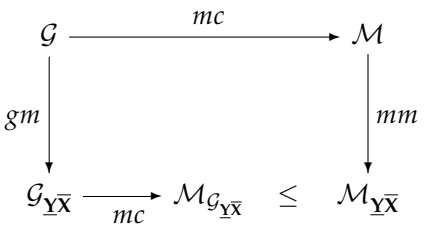
Now we have shown that for all $1 \leq i \leq k-1$, O_i is a parent of B in $\mathcal{M}$. It follows that X is adjacent to B, for otherwise $A \to B$ would be visible. Again, the edge is not $X \leftarrow B$ on pain of making $\mathcal{M}$ non-ancestral. So the edge between X and B in $\mathcal{M}$ is into B, but then there is an inducing path between X and Y relative to $\mathbf{L}$ in $\mathcal{G}$ that includes fewer observed variables than p does, which is a contradiction with our choice of p.

So our initial supposition is false. There is no inducing path between X and Y relative to $\mathbf{L}$ in $\mathcal{G}$, and hence X and Y are not adjacent in $\mathcal{M}_{\mathcal{G}}$.

Therefore $\mathcal{M} = \mathcal{M}_{\mathcal{G}}$. ■

Proof of Lemma 12

Proof Recall the diagram in Figure 7:



What we need to show is that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is an I-map of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, or in other words, whatever m-separation relation is true in the former is also true in the latter. To show this, it suffices to show that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is Markov equivalent to a supergraph of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$.

For that purpose, we first establish two facts: (1) every directed edge in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$ is also in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$; and (2) for every bi-directed edge $S \leftrightarrow T$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, S and T are also adjacent in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$; and the edge between S and T is either a bi-directed edge or an invisible directed edge in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$.

(1) is relatively easy to show. Note that for any $P \to Q$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, $P \notin \mathbf{Y}$, for otherwise P would not be an ancestor of Q in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, and hence would not be a parent of Q in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$; and likewise $Q \notin \mathbf{X}$, for otherwise Q would not be a descendant of P in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, and hence would not be a child of P in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$. Furthermore, because $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is a subgraph of $\mathcal{G}$, any inducing path between P and Q relative to $\mathbf{L}$ in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is also present in $\mathcal{G}$, and any directed path from P to Q in the former is also present in the latter. This entails that $P \to Q$ is also in $\mathcal{M}$, the MAG of $\mathcal{G}$. Since $P \notin \mathbf{Y}$ and $Q \notin \mathbf{X}$, $P \to Q$ is also present in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. So (1) is true.

(2) is less obvious. First of all, note that if $S \leftrightarrow T$ is in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, then there is an inducing path between S and T relative to $\mathbf{L}$ in $\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ that is into both S and T. This implies that $S, T \notin \mathbf{X}$, and moreover there is also an inducing path between S and T relative to $\mathbf{L}$ in $\mathcal{G}$ that is into both S and T. Hence there is an edge between S and T in $\mathcal{M}$, the MAG of $\mathcal{G}$. The edge in $\mathcal{M}$ is either $S \leftrightarrow T$ or, by Lemma 9, an invisible directed edge ($S \leftarrow T$ or $S \to T$).

Because $S, T \notin \mathbf{X}$, if $S \leftrightarrow T$ appears in $\mathcal{M}$, it also appears in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. If, on the other hand, the edge between S and T in $\mathcal{M}$ is directed, suppose without loss of generality that it is $S \to T$. Either $S \in \mathbf{Y}$, in which case we have $S \leftrightarrow T$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, because $S \to T$

is invisible in $\mathcal{M}$; or $S \notin \mathbf{Y}$, and $S \to T$ remains in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. In the latter case we need to show that $S \to T$ is still invisible in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. Suppose for the sake of contradiction that $S \to T$ is visible in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, that there is a vertex R not adjacent to T, such that either $R \ast\!\!\to S$ is in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ or there is a collider path c in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ between R and S that is into S on which every collider is a parent of T. We show that $S \to T$ is also visible in $\mathcal{M}$. Consider the two possible cases separately:

Case 1: $R \ast\!\!\to S$ is in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. If the edge is $R \to S$, it is also in $\mathcal{M}$, because manipulations of a MAG do not create new directed edges. We now show that R and T are not adjacent in $\mathcal{M}$. Suppose otherwise. The edge between R and T has to be $R \to T$ in $\mathcal{M}$. Note that $R \notin \mathbf{Y}$ for otherwise $R \to S$ would be deleted or changed into a bi-directed edge; and we have already shown that $T \notin \mathbf{X}$. It follows that $R \to T$ would be present in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ as well, a contradiction. Hence R and T are not adjacent in $\mathcal{M}$, and so the edge $S \to T$ is also visible in $\mathcal{M}$.

Suppose, on the other hand, the edge between R and S in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is $R \leftrightarrow S$. In $\mathcal{M}$ the edge is either (i) $R \leftrightarrow S$, or (ii) $R \to S$. (It can't be $R \leftarrow S$ because then $S \in \mathbf{Y}$ and the edge $S \to T$ would not remain in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$.)

If (i) is the case, we argue that R and T are not adjacent in $\mathcal{M}$. Since $R \leftrightarrow S \to T$ is in $\mathcal{M}$, if R and T are adjacent, it has to be $R \leftrightarrow T$ or $R \to T$. In the former case, $R \leftrightarrow T$ would still be present in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ (because obviously $R, T \notin \mathbf{X}$), which is a contradiction. In the latter case, $R \to T$ is invisible in $\mathcal{M}$, for otherwise it is easy to see that $S \to T$ would also be visible. So either $R \to T$ remains in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ (if $R \notin \mathbf{Y}$), or it turns into $R \leftrightarrow T$ (if $R \in \mathbf{Y}$). In either case R and T would still be adjacent in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, a contradiction. Hence R and T are not adjacent in $\mathcal{M}$, and so the edge $S \to T$ is also visible in $\mathcal{M}$.

If (ii) is the case, then either R and T are not adjacent in $\mathcal{M}$, in which case $S \to T$ is also visible in $\mathcal{M}$; or R and T are adjacent in $\mathcal{M}$, in which case we now show that $S \to T$ is still visible. The edge between R and T in $\mathcal{M}$ has to be $R \to T$ (in view of $R \to S \to T$). Since R and T are not adjacent in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, and $R \to S$ is turned into $R \leftrightarrow S$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, by the definition of lower-manipulation (Definition 11), $R \to T$ is visible but $R \to S$ is invisible in $\mathcal{M}$. Because $R \to T$ is visible, by definition, there is a vertex Q not adjacent to T such that $Q \ast\!\!\to R$ is in $\mathcal{M}$ or there is a collider path in $\mathcal{M}$ between Q and R that is into R on which every collider is a parent of T. But $R \to S$ is not visible, from which we can derive that $S \to T$ is visible in $\mathcal{M}$. Here is a sketch of the argument. If $Q \ast\!\!\to R$ is in $\mathcal{M}$, then Q and S must be adjacent (since otherwise $R \to S$ would be visible). It is then easy to derive that the edge between Q and S must be into S, which makes $S \to T$ visible. On the other hand, suppose there is a collider path c into R on which every collider is a parent of T. Then if there is a collider P on c such that $P \leftrightarrow S$ is in $\mathcal{M}$, we immediately get a collider path between Q and S that is into S on which every collider is a parent of T. This path makes $S \to T$ visible. Finally, if no collider on the path is a spouse of S, it is not hard to show that in order for $R \to S$ to be invisible, there has to be an edge between Q and S that is into S, which again makes $S \to T$ visible.

Case 2: There is a collider path c in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ between R and S that is into S on which every collider is a parent of T. We claim that every arrowhead on c, except possibly one at R, is also in $\mathcal{M}$. Because if an arrowhead is added at a vertex Q (which could be S) on c by the lower-manipulation, then $Q \in \mathbf{Y}$, but then the edge $Q \to T$ would not remain in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, a contradiction. So c is also a collider path in $\mathcal{M}$ that is into S. Furthermore, no new directed edges are introduced by lower-manipulation or upper-manipulation, so every collider on c is still a parent of T in $\mathcal{M}$.

It follows that if R and T are not adjacent in $\mathcal{M}$, then $S \to T$ is visible in $\mathcal{M}$. On the other hand, if R and T are adjacent in $\mathcal{M}$, it is either $R \leftrightarrow T$ or $R \to T$. Note that this edge is deleted in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. This implies that it is not $R \leftrightarrow T$ in $\mathcal{M}$: otherwise, the edge incident to R on c has to be bi-directed as well (since otherwise $\mathcal{M}$ is not ancestral), and hence if $R \leftrightarrow T$ is deleted, either the edge incident to R on c or the edge $S \to T$ should be deleted in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, which is a contradiction. So the edge is $R \to T$ in $\mathcal{M}$. Since $T \notin \mathbf{X}$ (for otherwise $S \to T$ would be deleted), $R \in \mathbf{Y}$, and $R \to T$ is visible in $\mathcal{M}$. But then it is easy to see that $S \to T$ is also visible in $\mathcal{M}$.

To summarize, we have shown that if $S \to T$ is visible in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, it is also visible in $\mathcal{M}$. Since it is not visible in $\mathcal{M}$, it is invisible in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ as well. Thus the edge between S and T is either a bi-directed edge or an invisible directed edge in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. Hence we have established (2).

The strategy to complete the proof is to show that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ can be transformed into a supergraph of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$ via a sequence of equivalence-preserving mark changes (Zhang and Spirtes, 2005; Tian, 2005). By (1) and (2), if $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is not yet a supergraph of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, it is because some bi-directed edges in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$ correspond to directed edges in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. For any such directed edge $P \to Q$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ (with $P \leftrightarrow Q$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$), (2) implies that $P \to Q$ is invisible. It is then not hard to check that conditions in Lemma 1 of Zhang and Spirtes (2005)[22] hold for $P \to Q$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, and thus it can be changed into $P \leftrightarrow Q$ while preserving Markov equivalence. Furthermore, making this change will not make any other such directed edge in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ visible. It follows that $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ can be transformed into a Markov equivalent graph that is a supergraph of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$. (We skip the details as they involve conditions for Markov equivalence we didn't have enough space to cover.)

Denote the supergraph by $\mathcal{I}$. It follows that if there is an m-connecting path between A and B given $\mathbf{C}$ in $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$, the path is also m-connecting in $\mathcal{I}$, the supergraph of $\mathcal{M}_{\mathcal{G}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}}$. Because $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ and $\mathcal{I}$ are Markov equivalent, there is also an m-connecting path between A and B given $\mathbf{C}$ in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. ■

Proof of Lemma 16

Proof It is not hard to check that for any two variables $P, Q \in \mathbf{O}$, if P and Q are adjacent in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, then they are adjacent in $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ (though the converse is not necessarily true, because an edge not definitely visible in $\mathcal{P}$ may still be visible in $\mathcal{M}$). Furthermore, when they are adjacent in both $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ and $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$, every non-circle mark on the edge in $\mathcal{P}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$ is "sound" in that the mark also appears in $\mathcal{M}_{\underline{\mathbf{Y}}\overline{\mathbf{X}}}$. The lemma obviously follows. ■

Proof of Lemma 21

22. Here is the Lemma: Let $\mathcal{M}$ be a MAG, and $A \to B$ a directed edge in $\mathcal{M}$. Let $\mathcal{M}'$ be the graph identical to $\mathcal{M}$ except that the edge between A and B is $A \leftrightarrow B$ in $\mathcal{M}'$. (In other words, $\mathcal{M}'$ is the result of simply changing $A \to B$ into $A \leftrightarrow B$ in $\mathcal{M}$.) $\mathcal{M}'$ is a MAG and Markov equivalent to $\mathcal{M}$ if and only if
(t1) there is no directed path from A to B other than $A \to B$ in $\mathcal{M}$;
(t2)] For every $C \to A$ in $\mathcal{M}$, $C \to B$ is also in $\mathcal{M}$; and for every $D \leftrightarrow A$ in $\mathcal{M}$, either $D \to B$ or $D \leftrightarrow B$ is in $\mathcal{M}$; and
(t3) there is no discriminating path for A on which B is the endpoint adjacent to A in $\mathcal{M}$.

Proof Spirtes and Richardson (1996), in proving their Lemma 18, gave a construction of an m-connecting path in $\mathcal{M}$ from a d-connecting path in $\mathcal{G}$. We describe the construction below.[23]

Let p be a minimal d-connecting path between A and B relative to $\mathbf{C}$ in $\mathcal{G}$ that is into A, minimal in the sense that no other d-connecting path between A and B relative to $\mathbf{C}$ that is into A is composed of fewer variables than p is.[24] Construct a sequence of variables in $\mathbf{O}$ in three steps.

Step 1: Form a sequence $\mathbf{T}$ of variables on p as follows. $\mathbf{T}[0] = A$, and $\mathbf{T}[n+1]$ is chosen to be the first vertex after $\mathbf{T}[n]$ on p that is either in $\mathbf{O}$ or a (latent) collider on p, until B is included in $\mathbf{T}$.

Step 2: Form a sequence $\mathbf{S}_0$ of variables in $\mathbf{O}$ of the same length as $\mathbf{T}$, which we assume contains m variables. For each $0 \leq n \leq m-1$, if $\mathbf{T}[n]$ is in $\mathbf{O}$, then $\mathbf{S}_0[n] = \mathbf{T}[n]$; otherwise $\mathbf{T}[n]$ is a (latent) collider on p, which, by the fact that p is d-connecting given $\mathbf{C}$, implies that there is a directed path from $\mathbf{T}[n]$ to a member of $\mathbf{C}$. So in this case, $\mathbf{S}_0[n]$ is chosen to be the first observed variable on a directed path from $\mathbf{T}[n]$ to a member of $\mathbf{C}$.

Step 3: Run the following iterative procedure:

k:=0

Repeat

If in $\mathbf{S}_k$ there is a triple of vertices $\langle \mathbf{S}_k[i-1], \mathbf{S}_k[i], \mathbf{S}_k[i+1] \rangle$ such that (1) there is an inducing path between $\mathbf{S}_k[i-1]$ and $\mathbf{S}_k[i]$ relative to $\mathbf{L}$ in $\mathcal{G}$ that is into $\mathbf{S}_k[i]$; (2) there is an inducing path between $\mathbf{S}_k[i]$ and $\mathbf{S}_k[i+1]$ relative to $\mathbf{L}$ in $\mathcal{G}$ that is into $\mathbf{S}_k[i]$; and (3) $\mathbf{S}_k[i]$ is in $\mathbf{C}$ and is an ancestor of either $\mathbf{S}_k[i-1]$ or $\mathbf{S}_k[i+1]$; then let sequence $\mathbf{S}_{k+1}$ be $\mathbf{S}_k$ with $\mathbf{S}_k[i]$ being removed;

k := k+1

Until there is no such triple of vertices in the sequence $\mathbf{S}_k$.

Let $\mathbf{S}_K$ denote the final outcome of the above three steps. Spirtes and Richardson (1996), in their Lemma 18, showed that $\mathbf{S}_K$ constitutes an m-connecting path between A and B relative to $\mathbf{C}$ in $\mathcal{M}$. We refer the reader to their paper for the detailed proof of this fact. What is left for us to show here is that the path constituted by $\mathbf{S}_K$ in $\mathcal{M}$ is either into A or out of A with an invisible edge.

In other words, we need to show that if the edge between $A = \mathbf{S}_K[0]$ and $\mathbf{S}_K[1]$ in $\mathcal{M}$ is $A \rightarrow \mathbf{S}_K[1]$, then this edge is invisible. Given Lemma 9, it suffices to show that there is an inducing path between A and $\mathbf{S}_K[1]$ relative to $\mathbf{L}$ in $\mathcal{G}$ that is into A. This is not hard to show. In fact, we can show by induction that for all $0 \leq k \leq K$, there is in $\mathcal{G}$ an inducing path between A and $\mathbf{S}_k[1]$ relative to $\mathbf{L}$ that is into A.

In the base case, notice that either (i) $\mathbf{S}_0[1]$ is an observed variable on p such that every variable between A and $\mathbf{S}_0[1]$ on p, if any, belongs to $\mathbf{L}$ and is a non-collider on p, or (ii) $\mathbf{S}_0[1]$ is the first observed variable on a directed path d starting from $\mathbf{T}[1]$ such that $\mathbf{T}[1]$ belongs to $\mathbf{L}$, lies on p and every variable between A and $\mathbf{T}[1]$ on p, if any, belongs to $\mathbf{L}$ and is a non-collider on p. In case (i), $p(A, \mathbf{S}_0[1])$ is an inducing path relative to $\mathbf{L}$, which is into A, because p is into A. In case (ii), consider $p(A, \mathbf{T}[1])$ and $d(\mathbf{T}[1], \mathbf{S}_0[1])$.

23. Their lemma addresses the more general case in which there may also be selection variables. The construction given here is an adaptation of theirs to fit our case.
24. In Spirtes and Richardson (1996), minimality means more than that the d-connecting path is a shortest one, but for this proof one only need to choose a shortest path.

Let W be the variable nearest to A on $p(A, \mathbf{T}[1])$ that is also on $d(\mathbf{T}[1], \mathbf{S}_0[1])$. ($W$ exists because $p(A, \mathbf{T}[1])$ and $d(\mathbf{T}[1], \mathbf{S}_0[1])$ at least intersect at $\mathbf{T}[1]$.) Then it is easy to see that a concatenation of $p(A, W)$ and $d(W, \mathbf{S}_0[1])$ forms an inducing path between A and $\mathbf{S}_0[1]$ relative to $\mathbf{L}$ in $\mathcal{G}$, which is into A because p is into A.

Now the inductive step. Suppose there is in $\mathcal{G}$ an inducing path between A and $\mathbf{S}_k[1]$ relative to $\mathbf{L}$ that is into A. Consider $\mathbf{S}_{k+1}[1]$. If $\mathbf{S}_{k+1}[1] = \mathbf{S}_k[1]$, it is trivial that there is an inducing path between A and $\mathbf{S}_{k+1}[1]$ that is into A. Otherwise, $\mathbf{S}_k[1]$ was removed in forming $\mathbf{S}_{k+1}$. But given the three conditions for removing $\mathbf{S}_k[1]$ in *Step 3* above, we can apply Lemma 32 (together with the inductive hypothesis) to conclude that there is an inducing path between A and $\mathbf{S}_{k+1}[1] = \mathbf{S}_k[2]$ relative to $\mathbf{L}$ in $\mathcal{G}$ that is into A. This concludes our argument. ■

Proof of Lemma 22

Proof This lemma is fairly obvious given Lemma 10. Let u be the path m-connecting A and B given $\mathbf{C}$ in $\mathcal{M}$. Let D (which could be B) be the vertex next to A on u. Construct a DAG $\mathcal{G}$ from $\mathcal{M}$ in the usual way: keep all the directed edges, replacing each bi-directed edge $X \leftrightarrow Y$ with $X \leftarrow L_{XY} \rightarrow Y$. Furthermore, if the edge between A and D is $A \rightarrow D$, it is invisible, so we can add $A \leftarrow L_{AD} \rightarrow D$ to the DAG. Then $\mathcal{G}$ is a DAG represented by $\mathcal{M}$. It is easy to check that there is a d-connecting path in $\mathcal{G}$ between A and B given $\mathbf{C}$ that is into A. ■

Proof of Lemma 23

Proof Note that because A is not an ancestor of any member of $\mathbf{C}$, if there is a path out of A d-connecting A and B given $\mathbf{C}$ in $\mathcal{G}$, the path must be a directed path from A to B. For otherwise there would be a collider on the path that is also a descendant of A, which implies that A is an ancestor of some member of $\mathbf{C}$. The sub-sequence of that path consisting of observed variables then constitutes a directed path from A to B in $\mathcal{M}$, which is of course out of A and also m-connecting given $\mathbf{C}$ in $\mathcal{M}$. The converse is as easy to show. ■

Proof of Lemma 27

Proof A path definitely m-connecting A and B given $\mathbf{C}$ in $\mathcal{P}$ is m-connecting in every MAG represented by $\mathcal{P}$, which is an immediate consequence of the definition of PAG. Let D be the vertex next to A on the definite m-connecting path in $\mathcal{P}$ between A and B given $\mathbf{C}$. All we need to show is that if the edge between A and D is not a definitely visible edge $A \rightarrow D$ in $\mathcal{P}$, then there exists a MAG represented by $\mathcal{P}$ in which the edge between A and D is not a visible edge out of A.

Obviously if the edge in $\mathcal{P}$ is not $A \rightarrow D$, there exists a MAG represented in $\mathcal{P}$ in which the edge is not $A \rightarrow D$, which follows from the fact that $\mathcal{P}$, by definition, displays all edge marks that are shared by all MAGs in the equivalence class.

So we only need to consider the case where the edge in $\mathcal{P}$ is $A \rightarrow D$, but it is not definitely visible. Now we need to use a fact proved in Lemma 3.3.4 of Zhang (2006, pp. 80): that we can turn $\mathcal{P}$ into a MAG by first changing every $\circ\!\!\rightarrow$ edge in $\mathcal{P}$ into a directed edge $\rightarrow$, and then orienting the circle component—the subgraph of $\mathcal{P}$ that consists of $\circ\!\!-\!\!\circ$ edges—into a DAG with no unshielded colliders.[25] Moreover, it is not

25. A triple of vertices $\langle X, Y, Z \rangle$ in a graph is called an *unshielded* triple if there is an edge between X and Y, an edge between Y and Z, but no edge between X and Z. It is an *unshielded collider* if both the edge between X and Y and the edge between Z and Y are into Y.

hard to show, using the result in Meek (1995a), that we can orient the circle component—a chordal graph—into a DAG free of unshielded colliders in which every edge incident to A is oriented out of A.

Let the resulting MAG be $\mathcal{M}$. We show that $A \rightarrow D$ is invisible in $\mathcal{M}$. Suppose for the sake of contradiction that it is visible in $\mathcal{M}$. Then there exists in $\mathcal{M}$ a vertex E not adjacent to D such that either $E* \rightarrow A$ or there is a collider path between E and A that is into A and every collider on the path is a parent of D. In the former case, since $A \rightarrow D$ is not definitely visible in $\mathcal{P}$, the edge between E and A is not into A in $\mathcal{P}$, but then that edge will not be oriented as into A by our construction of $\mathcal{M}$. So the former case is impossible.

In the latter case, denote the collider path by $\langle E, E_1, ..., E_m, A \rangle$. Obviously every edge on $\langle E_1, ..., E_m, A \rangle$ is bi-directed, and so also occurs in $\mathcal{P}$ (because our construction of $\mathcal{M}$ does not introduce extra bi-directed edges). There are then two cases to consider:

Case 1: The edge between E and E_1 is also into E_1 in $\mathcal{P}$. Then the collider path appears in $\mathcal{P}$. We don't have space to go into the details here, but there is an orientation rule in constructing PAGs that makes use of a construct called "discriminating path" (e.g., Spirtes et al., 1999; Zhang, forthcoming), which would imply that if the collider path appears in $\mathcal{P}$, and every E_i ($1 \leq i \leq m$) is a parent of D in a representative MAG $\mathcal{M}$, then every E_i is also a parent of D in $\mathcal{P}$. It follows that $A \rightarrow D$ is definitely visible in $\mathcal{P}$, a contradiction.

Case 2: The edge between E and E_1 is not into E_1 in $\mathcal{P}$, but is oriented as into E_1 in $\mathcal{M}$. This is possible only if the edge is $E \circ\!\!-\!\!\circ E_1$ in $\mathcal{P}$. But we also have $E_1 \leftrightarrow E_2$ (E_2 could be A) in $\mathcal{P}$, which, by Lemma 3.3.1 in Zhang (2006, pp. 77), implies that $E \leftrightarrow E_2$ is in $\mathcal{P}$. Then $\langle E, E_2, \ldots, A \rangle$ makes $A \rightarrow D$ definitely visible in $\mathcal{P}$, which is a contradiction. ■

Proof of Lemma 28

Proof Note that since A does not have a descendant in $\mathbf{C}$, an m-connecting path out of A given $\mathbf{C}$ in $\mathcal{M}$ has to be a directed path from A to B such that every vertex on the path is not in $\mathbf{C}$. Then a shortest such path has to be uncovered,[26] and so will correspond to a definite m-connecting path between A and B given $\mathbf{C}$ in $\mathcal{P}$ (on which every vertex is a definite non-collider). This path is not into A in $\mathcal{P}$ because $\mathcal{P}$ is the PAG for $\mathcal{M}$ in which the path is out of A. ■

Proof of Lemma 29

Proof Let D be the vertex next to A on the definite m-connecting path in $\mathcal{P}$. Since the edge between A and D is not into A in $\mathcal{P}$, there exists a MAG represented by $\mathcal{P}$ in which the edge is out of A (which follows from the definition of PAG). Such a MAG obviously satisfies the lemma. ■

References

R.A. Ali, T. Richardson, and P. Spirtes. Markov equivalence for ancestral graphs. Technical Report 466, Department of Statistics, University of Washington, 2004.

26. A path is called *uncovered* if every consecutive triple on the path is unshielded (cf. Footnote 25).

S. Andersson, D. Madigan, and M. Pearlman. A characterization of Markov equivalence classes of acyclic digraphs. *The Annals of Statistics* 25(2):505-541, 1997.

D.M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87-98, Morgan Kaufmann, 1995.

D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research* 3:507-554, 2002.

D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks 20*, pages 507-534, 1990.

Y. Huang and M. Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of 22nd Conference on Uncertainty in Artificial Intelligence*, pages 217-224, AUAI Press, 2006.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403-411, Morgan Kaufmann, 1995a.

C. Meek. Strong completeness and faithfulness in Bayesian networks, In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 411-418, Morgan Kaufmann, 1995b.

J. Pearl. Causal diagrams for empirical research. *Biometrika* 82:669-710, 1995.

J. Pearl. Graphs, causality and structural equation models. *Sociological Methods and Research* 27:226-284, 1998.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.

J.W. Pratt and R. Schlaifer. On the interpretation and observation of laws. *Journal of Econometrics* 39:23-52, 1988.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *The Annals of Statistics* 30(4):962-1030, 2002.

T. Richardson and P. Spirtes. Causal inference via ancestral graph models. In P. Green, N. Hjort, and S. Richardson, editors, *Highly Structured Stochastic Systems*. Oxford University Press, USA, 2003.

J. Robins. A new approach to causal inference in mortality studies with sustained exposure periods—applications to control of the healthy worker survivor effect. *Mathematical Modeling* 7:1393-1512, 1986.

S. Shimizu, P.O. Hoyer, A. Hyvarinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* 7:2003-30, 2006.

I. Shpitser and J. Pearl. Identification of conditional interventional distributions. In *Proceedings of 22nd Conference on Uncertainty in Artificial Intelligence*, pages 437-444, AUAI Press, 2006.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Springer-Verlag., New York, 1993. (2nd ed., MIT Press, Cambridge, MA, 2000.)

P. Spirtes, C. Meek, and T. Richardson. An algorithm for causal inference in the presence of latent variables and selection bias. In C. Glymour and G.F. Cooper, editors, *Computation, Causation, and Discovery*. MIT Press, Cambridge, MA, 1999.

P. Spirtes and T. Richardson. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, 1996. URL http://citeseer.ist.psu.edu/spirtes97polynomial.html.

P. Spirtes and T. Verma. Equivalence of causal models with latent variables. Technical Report Phil-36, Department of Philosophy, Carnegie Mellon University, 1992.

J. Tian and J. Pearl. On the identification of causal effects. Technical Report, Department of Computer Science, Iowa State University, 2004.

J. Tian. Generating Markov equivalent maximal ancestral graphs by single edge replacement. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 591-598, AUAI Press, 2005.

C. Winship and L.S. Morgan. The estimation of causal effects from observational data. *Annual Review of Sociology* 25:659-706, 1999.

J. Zhang and P.Spirtes. A transformational characterization of Markov equivalence for directed acyclic graphs with latent variables. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 667-674, AUAI Press, 2005.

J. Zhang. *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD dissertation, Department of Philosophy, Carnegie Mellon University, 2006. URL www.hss.caltech.edu/~jiji/dissertation.pdf.

J. Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, forthcoming.

H. Zhao, Z. Zheng, and B. Liu. On the Markov equivalence of maximal ancestral graphs. *Science in China (Mathematics)*, 48(4):548-562, 2005.

JMLR: Workshop and Conference Proceedings 9:1941–1979, 2008 Submitted 6/07; Revised 2/08; Published 9/08

Complete Identification Methods for the Causal Hierarchy

Ilya Shpitser ILYAS@CS.UCLA.EDU
Judea Pearl JUDEA@CS.UCLA.EDU
Cognitive Systems Laboratory
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Editor: Peter Spirtes

Abstract

We consider a hierarchy of queries about causal relationships in graphical models, where each level in the hierarchy requires more detailed information than the one below. The hierarchy consists of three levels: associative relationships, derived from a joint distribution over the observable variables; cause-effect relationships, derived from distributions resulting from external interventions; and counterfactuals, derived from distributions that span multiple "parallel worlds" and resulting from simultaneous, possibly conflicting observations and interventions. We completely characterize cases where a given causal query can be computed from information lower in the hierarchy, and provide algorithms that accomplish this computation. Specifically, we show when effects of interventions can be computed from observational studies, and when probabilities of counterfactuals can be computed from experimental studies. We also provide a graphical characterization of those queries which cannot be computed (by any method) from queries at a lower layer of the hierarchy.

Keywords: causality, graphical causal models, identification

1. Introduction

The human mind sees the world in terms of causes and effects. Understanding and mastering our environment hinges on answering questions about cause-effect relationships. In this paper we consider three distinct classes of causal questions forming a hierarchy.

The first class of questions involves associative relationships in domains with uncertainty, for example, "I took an aspirin after dinner, will I wake up with a headache?" The tools needed to formalize and answer such questions are the subject of probability theory and statistics, for they require computing or estimating some aspects of a joint probability distribution. In our aspirin example, this requires estimating the conditional probability $P(headache \mid aspirin)$ in a population that resembles the subject in question, that is, sharing age, sex, eating habits and any other traits that can be measured. Associational relationships, as is well known, are insufficient for establishing causation. We nevertheless place associative questions at the base of our causal hierarchy, because the probabilistic tools developed in studying such questions are instrumental for computing more informative causal queries, and serve therefore as an easily available starting point from which such computations can begin.

The second class of questions involves responses of outcomes of interest to outside interventions, for instance, "if I take an aspirin now, will I wake up with a headache?" Questions of this type are normally referred to as *causal effects*, sometimes written as $P(headache \mid do(aspirin))$. They differ, of course from the associational counterpart $P(headache \mid aspirin)$, because all mechanisms which normally determine aspirin taking behavior, for example, taste of aspirin, family advice, time pressure, etc. are irrelevant in evaluating the effect of a new decision.

To estimate effects, scientists normally perform randomized experiments where a sample of units drawn from the population of interest is subjected to the specified manipulation directly. In our aspirin example, this might involve treating a group of subjects with aspirin and comparing their response to untreated subjects, both groups being selected at random from a population resembling the decision maker in question. In many cases, however, such a direct approach is not possible due to expense or ethical considerations. Instead, investigators have to rely on observational studies to infer effects. A fundamental question in causal analysis is to determine when effects can be inferred from statistical information, encoded as a joint probability distribution, obtained under normal, intervention-free behavior. A key point here is that in order to make causal inferences from statistics, additional causal assumptions are needed. This is because without any assumptions it is possible to construct multiple "causal stories" which can disagree wildly on what effect a given intervention can have, but agree precisely on all observables. For instance, smoking may be highly correlated with lung cancer either because it causes lung cancer, or because people who are genetically predisposed to smoke may also have a gene responsible for a higher cancer incidence rate. In the latter case there will be no effect of smoking on cancer. Distinguishing between such causal stories requires additional, non-statistical language. In this paper, the language that we use for this purpose is the language of graphs, and our causal assumptions will be encoded by a special directed graph called a *causal diagram*.

The use of directed graphs to represent causality is a natural idea that arose multiple times independently: in genetics (Wright, 1921), econometrics (Haavelmo, 1943), and artificial intelligence (Pearl, 1988; Spirtes et al., 1993; Pearl, 2000). A causal diagram encodes variables of interest as nodes, and possible direct causal influences between two variables as arrows. Associated with each node in a causal diagram is a stable causal mechanism which determines its value in terms of the values of its parents. Unlike Bayesian networks (Pearl, 1988), the relationships between variables are assumed to be deterministic and uncertainty arises due to the presence of unobserved variables which have influence on our domain.

The first question we consider is under what conditions the effect of a given intervention can be computed from just the joint distribution over observable variables, which is obtainable by statistical means, and the causal diagram, which is either provided by a human expert, or inferred from experimental studies. This *identification problem* has received consideration attention in the statistics, epidemiology, and causal inference communities (Pearl, 1993a; Spirtes et al., 1993; Pearl and Robins, 1995; Pearl, 1995; Kuroki and Miyakawa, 1999; Pearl, 2000). In the subsequent sections, we solve the identification problem for causal effects by providing a graphical characterization for all non-identifiable effects, and an algorithm for computing all identifiable effects. Note that this identification problem actually involves two "worlds:" the original world where no interventions took place furnishes us with a probability distribution from which to make inferences about the second, post-intervention world. The crucial feature

of causal effect queries which distinguishes them from more complex questions in our hierarchy is that they are restricted to the post-intervention world alone.

The third and final class of queries we consider are *counterfactual* or "what-if" questions which arise when we simultaneously ask about multiple hypothetical worlds, with potentially conflicting interventions or observations. An example of such a question would be "I took an aspirin, and my headache is gone; would I have had a headache had I not taken that aspirin?" Unlike questions involving interventions, counterfactuals contain conflicting information: in one world aspirin was taken, in another it was not. It is unclear therefore how to set up an effective experimental procedure for evaluating counterfactuals, let alone how to compute counterfactuals from observations alone. If everything about our causal domain is known, in other words if we have knowledge of both the causal mechanisms and the distributions over unobservable variables, it is possible to compute counterfactual questions directly (Balke and Pearl, 1994b). However, knowledge of precise causal mechanisms is not generally available, and the very nature of unobserved variables means their stochastic behavior cannot be estimated directly. We therefore consider the more practical question of how to compute counterfactual questions from both experimental studies and the structure of the causal diagram.

It may seem strange, in light of what we said earlier about the difficulty of conducting experimental studies, that we take such studies as given. It is nevertheless important that we understand when it is that "what-if" questions involving multiple worlds can be inferred from quantities computable in one world. Our hierarchical approach to identification allows us to cleanly separate difficulties that arise due to multiplicity of worlds from those involved in the identification of causal effects. We provide a complete solution to this version of the identification problem by giving algorithms which compute identifiable counterfactuals from experimental studies, and provide graphical conditions for the class of non-identifiable counterfactuals, where our algorithms fail. Our results can, of course, be combined to give conditions where counterfactuals can be computed from observational studies.

The paper is organized as follows. Section 2 introduces the notation and mathematical machinery needed for causal analysis. Section 3 considers the problem of identifying causal effects from observational studies. Section 4 considers identification of counterfactual queries, while Section 5 summarizes the conclusions. Most of the proofs are deferred to the appendix. This paper consolidates and expands previous results (Shpitser and Pearl, 2006a,b, 2007). Some of the results found in this paper were also derived independently elsewhere (Huang and Valtorta, 2006b,a).

2. Notation and Definitions

The primary object of causal inquiry is a probabilistic causal model. We will denote variables by uppercase letters, and their values by lowercase letters. Similarly, sets of variables will be denoted by bold uppercase, and sets of values by bold lowercase.

Definition 1 *A probabilistic causal model (PCM) is a tuple* $M = \langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, P(\boldsymbol{u}) \rangle$, *where*

- $\boldsymbol{U}$ *is a set of background or exogenous variables, which cannot be observed or experimented on, but which affect the rest of the model.*
- $\boldsymbol{V}$ *is a set* $\{V_1, \ldots, V_n\}$ *of observable or endogenous variables. These variables are functionally dependent on some subset of* $\boldsymbol{U} \cup \boldsymbol{V}$.

- *$\boldsymbol{F}$ is a set of functions $\{f_1, \ldots, f_n\}$ such that each f_i is a mapping from a subset of $\boldsymbol{U} \cup \boldsymbol{V} \setminus \{V_i\}$ to V_i, and such that $\bigcup \boldsymbol{F}$ is a function from $\boldsymbol{U}$ to $\boldsymbol{V}$.*
- *$P(\boldsymbol{u})$ is a joint probability distribution over $\boldsymbol{U}$.*

The set of functions **F** in this definition corresponds to the causal mechanisms, while **U** represents the background context that influences the observable domain of discourse **V**, yet remains outside it. Our ignorance of the background context is represented by a distribution $P(\mathbf{u})$. This distribution, together with the mechanisms in **F**, induces a distribution $P(\mathbf{v})$ over the observable domain. The causal diagram, our vehicle for expressing causal assumptions, is defined by the causal model as follows. Each observable variable $V_i \in \mathbf{V}$ corresponds to a vertex in the graph. Any two variables $V_i \in \mathbf{U} \cup \mathbf{V}$, $V_j \in \mathbf{V}$ such that V_i appears in the description of f_j are connected by a directed arrow from V_i to V_j. Furthermore, we make two additional assumptions in this paper. The first is that $P(\mathbf{u}) = \prod_{u_i \in \mathbf{u}} P(u_i)$, and each $U_i \in \mathbf{U}$ is used in at most two functions in F.[1] The second is that all induced graphs must be acyclic. Models in which these two assumptions hold are called recursive semi-Markovian. A graph defined as above from a causal model M is said to be a causal diagram *induced* by M. Graphs induced by semi-Markovian models are themselves called semi-Markovian. Figures 1 and 2 show some examples of causal diagrams of recursive semi-Markovian models.

The functions in **F** are assumed to be *modular* in a sense that changes to one function do not affect any other. This assumption allows us to model how a PCM would react to changes imposed from the outside. The simplest change that is possible for causal mechanisms of a variable set **X** would be one that removes the mechanisms entirely and sets **X** to a specific value **x**. This change, denoted by $do(\mathbf{x})$ (Pearl, 2000), is called an *intervention*. An intervention $do(\mathbf{x})$ applied to a model M results in a *submodel* $M_{\mathbf{x}}$. The effects of interventions will be formulated in several ways. For any given **u**, the effect of $do(\mathbf{x})$ on a set of variables **Y** will be represented by *counterfactual variables* $Y_{\mathbf{x}}(\mathbf{u})$, where $Y \in \mathbf{Y}$. As **U** varies, the counterfactuals $Y_{\mathbf{x}}(\mathbf{u})$ will vary as well, and their *interventional distribution*, denoted by $P(\mathbf{y} \mid do(\mathbf{x}))$ or $P_{\mathbf{x}}(\mathbf{y})$ will be used to define the effect of **x** on **Y**. We will denote the event "variable Y attains value y in $M_{\mathbf{x}}$" by the shorthand $y_{\mathbf{x}}$.

Interventional distributions are a mathematical formalization of an intuitive notion of effect of action. We now define joint probabilities on counterfactuals, in multiple worlds, which will serve as the formalization of counterfactual queries. Consider a conjunction of events $\gamma = y^1_{\mathbf{x}^1} \wedge \ldots \wedge y^k_{\mathbf{x}^k}$. If all the subscripts $\mathbf{x}^i$ are the same and equal to **x**, γ is simply a set of assignments of values to variables in $M_{\mathbf{x}}$, and $P(\gamma) = P_{\mathbf{x}}(y^1, \ldots, y^k)$. However, if the actions $do(\mathbf{x}^i)$ are not the same, and potentially contradictory, a single submodel is no longer sufficient. Instead, γ is really invoking multiple causal worlds, each represented by a submodel $M_{\mathbf{x}^i}$. We assume each submodel shares the same set of exogenous variables **U**, corresponding to the shared causal context or background history of the hypothetical worlds. Because the submodels are linked by common context, they can really be considered as one large causal model, with its own induced graph, and joint distribution over observable variables. $P(\gamma)$ can then be defined as a marginal distribution in this causal model. Formally, $P(\gamma) = \sum_{\{\mathbf{u} | \mathbf{u} \models \gamma\}} P(\mathbf{u})$, where $\mathbf{u} \models \gamma$ is taken to mean that each variable assignment in γ holds true in the corresponding submodel of M when the exogenous variables **U** assume values **u**. In this way, $P(\mathbf{u})$

1. Our results are generalizable to other $P(\mathbf{u})$ distributions which may not have such a simple form, but which can be represented by a set of bidirected arcs in such a way that whenever two sets of **U** variables are d-separated from each other, they are marginally independent. However, the exact conditions under which this graphical representation is valid are beyond the scope of this paper.

induces a distribution on all possible counterfactual variables in M. In this paper, we will represent counterfactual utterances by joint distributions such as $P(\gamma)$ or conditional distributions such as $P(\gamma \mid \delta)$, where γ and δ are conjunctions of counterfactual events. Pearl (2000) discusses counterfactuals, and their probabilistic representation used in this paper in greater depth.

A fundamental question in causal inference is whether a given causal question, either interventional or counterfactual in nature, can be uniquely specified by the assumptions embodied in the causal diagram, and easily available information, usually statistical, associated with the causal model. To get a handle on this question, we introduce an important notion of *identifiability* (Pearl, 2000).

Definition 2 (identifiability) *Consider a class of models $\mathbf{M}$ with a description T, and objects ϕ and θ computable from each model. We say that ϕ is θ-identified in T if ϕ is uniquely computable from θ in any $M \in \mathbf{M}$. In this case all models in $\mathbf{M}$ which agree on θ will also agree on ϕ.*

If ϕ is θ-identifiable in T, we write $T, \theta \vdash_{id} \phi$. Otherwise, we write $T, \theta \nvdash_{id} \phi$. The above definition leads immediately to the following corollary which we will use to prove non-identifiability results.

Corollary 3 *Let T be a description of a class of models $\mathbf{M}$. Assume there exist $M^1, M^2 \in \mathbf{M}$ that share objects θ, while ϕ in M^1 is different from ϕ in M^2. Then $T, \theta \nvdash_{id} \phi$.*

In our context, the objects ϕ, θ are probability distributions derived from the PCM, where θ represents available information, while ϕ represents the quantity of interest. The description T is a specification of the properties shared all causal models under consideration, or, in other words, the set of assumptions we wish to impose on those models. Since we chose causal graphs as a language for specifying assumptions, T corresponds to a given graph.

Graphs earn their ubiquity as a specification language because they reflect in many ways the way people store experiential knowledge, especially cause-effect relationships. The ease with which people embrace graphical metaphors for causal and probabilistic notions—ancestry, neighborhood, flow, and so on—are proof of this affinity, and help ensure that the assumptions specified are meaningful and reliable. A consequence of this is that probabilistic dependencies among variables can be verified by checking if the flow of influence is blocked along paths linking the variables. By a path we mean a sequence of distinct nodes where each node is connected to the next in the sequence by an edge. The precise way in which the flow of dependence can be blocked is defined by the notion of d-separation (Pearl, 1986; Verma, 1986; Pearl, 1988). Here we generalize d-separation somewhat to account for the presence of bidirected arcs in causal diagrams.

Definition 4 (d-separation) *A path p in G is said to be d-separated by a set $\mathbf{Z}$ if and only if either*

1. *p contains one of the following three patterns of edges: $I \to M \to J$, $I \leftrightarrow M \to J$, or $I \leftarrow M \to J$, such that $M \in \mathbf{Z}$, or*
2. *p contains one of the following three patterns of edges: $I \to M \leftarrow J$, $I \leftrightarrow M \leftarrow J$, $I \leftrightarrow M \leftrightarrow J$, such that $De(M)_G \cap \mathbf{Z} = \emptyset$.*

Two sets $\mathbf{X}, \mathbf{Y}$ are said to be d-separated given $\mathbf{Z}$ in G if all paths from $\mathbf{X}$ to $\mathbf{Y}$ in G are d-separated by $\mathbf{Z}$. Paths or sets which are not d-separated are said to be d-connected. What allows us to connect this notion of blocking of paths in a causal diagram to the notion of probabilistic independence among variables is that the probability distribution over $\mathbf{V}$ and $\mathbf{U}$ in a causal model can be represented as a product of factors, such that each observable node has a factor corresponding to its conditional distribution given the values of its parents in the graph. In other words, $P(\mathbf{v}, \mathbf{u}) = \prod_i P(x_i \mid pa(x_i)_G)$.

Whenever the above factor decomposition holds for a distribution $P(\mathbf{v}, \mathbf{u})$ and a graph G, we say G is an I-map of $P(\mathbf{v}, \mathbf{u})$. The following theorem links d-separation of vertex sets in an I-map G with the independence of corresponding variable sets in P.

Theorem 5 *If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map. Furthermore, the causal diagram induced by any semi-Markovian PCM M is an I-map of the distribution $P(\boldsymbol{v}, \boldsymbol{u})$ induced by M.*

Note that it's easy to rephrase the above theorem in terms of ordinary directed acyclic graphs, since each semi-Markovian graph is really an abbreviation where each bidirected arc stands for two directed arcs emanating from a hidden common cause. We will abbreviate this statement of d-separation, and corresponding independence by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G$, following the notation of Dawid (1979). For example in the graph shown in Figure 6 (a), $X \not\perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y \mid Z$, while in Figure 6 (b), $X \perp\!\!\!\perp Y$ and $X \not\perp\!\!\!\perp Y \mid Z$.

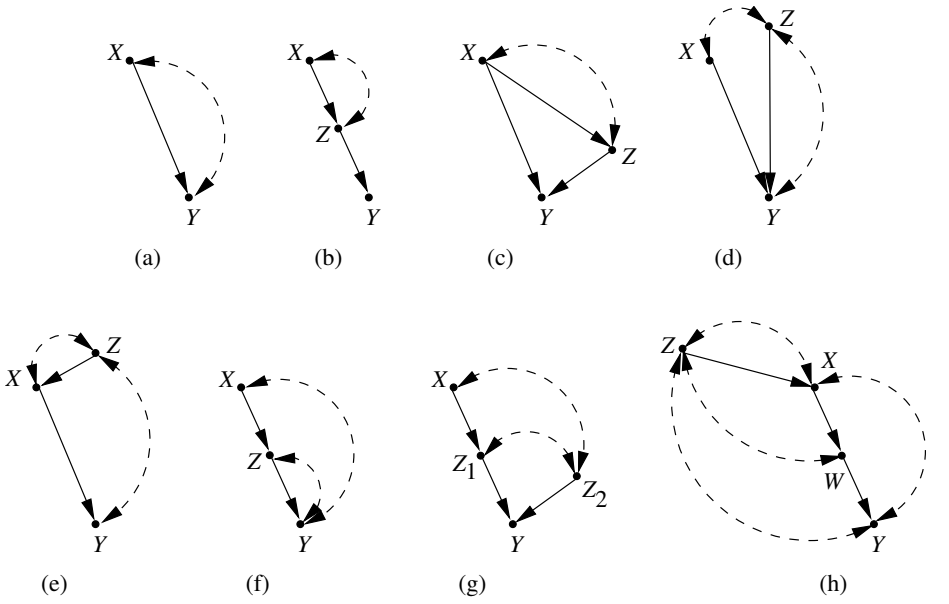


Figure 1: Causal graphs where $P(y \mid do(\mathbf{x}))$ is not identifiable

Finally we consider the axioms and inference rules we will need. Since PCMs contain probability distributions, the inference rules we would use to compute queries in PCMs would certainly include the standard axioms of probability. They also include a set of axioms which govern the behavior of counterfactuals, such as Effectiveness, Composition, etc. (Galles and Pearl, 1998; Halpern, 2000; Pearl, 2000). However, in

this paper, we will concentrate on a set of three identities applicable to interventional distributions known as do-calculus (Pearl, 1993b, 2000):

- Rule 1: $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{z}, \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{x}}}}$
- Rule 2: $P_{\mathbf{x},\mathbf{z}}(\mathbf{y} \mid \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{z}, \mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{x}},\underline{\mathbf{z}}}}$
- Rule 3: $P_{\mathbf{x},\mathbf{z}}(\mathbf{y} \mid \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{x}},\overline{z(\mathbf{w})}}}$

where $Z(\mathbf{W}) = \mathbf{Z} \setminus An(\mathbf{W})_{G_{\overline{\mathbf{X}}}}$, and $G_{\overline{\mathbf{x}},\underline{\mathbf{y}}}$ stands for a directed graph obtained from G by removing all incoming arrows to $\mathbf{X}$ and all outgoing arrows from $\mathbf{Y}$. The rules of do-calculus provide a way of linking ordinary statistical distributions with distributions resulting from various manipulations.

In the remainder of this section we will introduce relevant graphs and graph-theoretic terminology which we will use in the rest of the paper. First, having defined causal diagrams induced by natural causal models, we consider the graphs induced by models derived from interventional and counterfactual queries. We note that in a given submodel $M_{\mathbf{x}}$, the mechanisms determining $\mathbf{X}$ no longer make use of the parents of $\mathbf{X}$ to determine their values, but instead set them independently to constant values $\mathbf{x}$. This means that the induced graph of $M_{\mathbf{x}}$ derived from a model M inducing graph G can be obtained from G by removing all arrows incoming to $\mathbf{X}$, in other words $M_{\mathbf{x}}$ induces $G_{\overline{\mathbf{x}}}$. A counterfactual $\gamma = y^1_{\mathbf{x}^1} \wedge \ldots \wedge y^k_{\mathbf{x}^k}$, as we already discussed invokes multiple hypothetical causal worlds, each represented by a submodel, where all worlds share the same background context $\mathbf{U}$. A naive way to graphically represent these worlds would be to consider all the graphs $G_{\overline{\mathbf{X}^i}}$ and have them share the $\mathbf{U}$ nodes. It turns out this representation suffers from certain problems. In Section 4 we discuss this issue in more detail and suggest a more appropriate graphical representation of counterfactual situations.

We denote $Pa(.)_G, Ch(.)_G, An(.)_G, De(.)_G$ as the sets of parents, children, ancestors, and descendants of a given set in G. We denote $G_{\mathbf{X}}$ to be the subgraph of G containing all vertices in $\mathbf{X}$, and edges between these vertices, while the set of vertices in a given graph G is given by $\mathbf{ver}(G)$. As a shorthand, we denote $G_{\mathbf{ver}(G) \setminus \mathbf{ver}(G')}$ as $G \setminus G'$ or $G \setminus \mathbf{X}$, if $\mathbf{X} = \mathbf{ver}(G')$, and G' is a subgraph of G. We will call the set $\{X \in G \mid De(X)_G = \emptyset\}$ the *root set* of G. A path connecting X and Y which begins with an arrow pointing to X is called a *back-door path* from X, while a path beginning with an arrow pointing away from X is called a *front-door path* from X.

The goal of this paper is a complete characterization of causal graphs which permit the answering of causal queries of a given type. This characterization requires the introduction of certain key graph structures.

Definition 6 (tree) *A graph G such that each vertex has at most one child, and only one vertex (called the root) has no children is called a tree.*

Note that this definition reverses the usual direction of arrows in trees as they are generally understood in graph theory. If we ignore bidirected arcs, graphs in Figure 1 (a), (b), (d), (e), (f), (g), and (h) are trees.

Definition 7 (forest) *A graph G such that each vertex has at most one child is called a forest.*

Note that the above two definitions reverse the arrow directionality usual for these structures.

Definition 8 (confounded path) *A path where all directed arrowheads point at observable nodes, and never away from observable nodes is called a confounded path.*

The graph in Figure 1 (g) contains a confounded path from Z_1 to Z_2.

Definition 9 (c-component) *A graph G where any pair of observable nodes is connected by a confounded path is called a c-component (confounded component).*

Graphs in Figure 1 (a), (d), (e), (f), and (h) are c-components. Some graphs contain multiple c-components, for example the graph in Figure 1 (b) has two maximal c-components: $\{Y\}$, and $\{X, Z\}$. We will denote the set of maximal c-components of a given graph G by $C(G)$. The importance of c-components stems from the fact that that the observational distribution $P(\mathbf{v})$ can be expressed as a product of factors $P_{\mathbf{v}\setminus\mathbf{s}}(\mathbf{s})$, where each $\mathbf{s}$ is a set of nodes forming a c-component. This important property is known as *c-component factorization*, and we will this property extensively in the remainder of the manuscript to decompose identification problems into smaller subproblems.

In the following sections, we will show how the graph structures we defined in this section are key for characterizing cases when $P_{\mathbf{x}}(\mathbf{y})$ and $P(\gamma)$ can be identified from available information.

3. Identification of Causal Effects

Like probabilistic dependence, the notion of causal effect of X on Y has an interpretation in terms of flow. Intuitively, X has an effect on Y if changing X causes Y to change. Since intervening on X cuts off X from the normal causal influences of its parents in the graph, we can interpret the causal effect of X on Y as the flow of dependence which leaves X via outgoing arrows only.

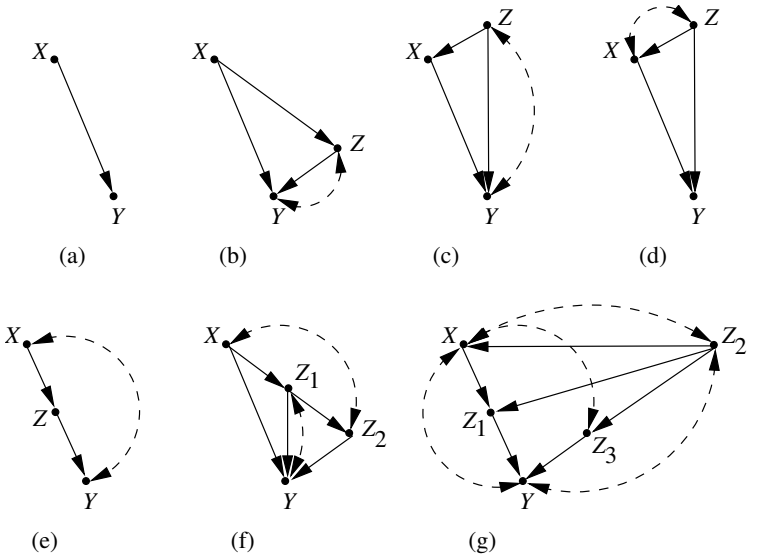


Figure 2: Causal graphs admitting identifiable effect $P(y \mid do(x))$

Recall that our ultimate goal is to express distributions of the form $P(\mathbf{y} \mid do(\mathbf{x}))$ in terms of the joint distribution $P(\mathbf{v})$. The interpretation of effect as downward dependence immediately suggests a set of graphs where this is possible. Specifically, whenever all d-connected paths from $\mathbf{X}$ to $\mathbf{Y}$ are front-door from $\mathbf{X}$, the causal effect $P(\mathbf{y} \mid do(\mathbf{x}))$ is equal to $P(\mathbf{y} \mid \mathbf{x})$. In graphs shown in Figure 2 (a) and (b) causal effect $P(y \mid do(x))$ has this property.

In general, we don't expect acting on $\mathbf{X}$ to produce the same effect as observing $\mathbf{X}$ due to the presence of back-door paths between $\mathbf{X}$ and $\mathbf{Y}$. However, d-separation gives us a way to block undesirable paths by conditioning. If we can find a set $\mathbf{Z}$ that blocks all back-door paths from $\mathbf{X}$ to $\mathbf{Y}$, we obtain the following: $P(\mathbf{y} \mid do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{y} \mid \mathbf{z}, do(\mathbf{x}))P(\mathbf{z} \mid do(\mathbf{x}))$. The term $P(\mathbf{y} \mid \mathbf{z}, do(\mathbf{x}))$ is reduced to $P(\mathbf{y} \mid \mathbf{z}, \mathbf{x})$ since the influence flow from $\mathbf{X}$ to $\mathbf{Y}$ is blocked by $\mathbf{Z}$. However, the act of adjusting for $\mathbf{Z}$ introduced a new effect we must compute, corresponding to the term $P(\mathbf{z} \mid do(\mathbf{x}))$. If it so happens that no variable in $\mathbf{Z}$ is a descendant of $\mathbf{X}$, we can reduce this term to $P(\mathbf{z})$ using the intuitive argument that acting on effects should not influence causes, or a more formal appeal to rule 3 of do-calculus. Computing effects in this way is always possible if we can find a set $\mathbf{Z}$ blocking all back-door paths which contains no descendants of $\mathbf{X}$. This is known as the *back-door criterion* (Pearl, 1993a, 2000). Figs. 2 (c) and (d) show some graphs where the node z satisfies the back-door criterion with respect to $P(y \mid do(x))$, which means $P(y \mid do(x))$ is identifiable.

The back-door criterion can fail—a common way involves a confounder that is unobserved, which prevents adjusting for it. Surprisingly, it is sometimes possible to identify the effect of $\mathbf{X}$ on $\mathbf{Y}$ even in the presence of such a confounder. To do so, we want to find a set $\mathbf{Z}$ located downstream of $\mathbf{X}$ but upstream of $\mathbf{Y}$, such that the downward flow of the effect of $\mathbf{X}$ on $\mathbf{Y}$ can be decomposed into the flow from $\mathbf{X}$ to $\mathbf{Z}$, and the flow from $\mathbf{Z}$ to $\mathbf{Y}$. Clearly, in order for this to happen $\mathbf{Z}$ must d-separate all front-door paths from $\mathbf{X}$ to $\mathbf{Y}$. However, in order to make sure that the component effects $P(\mathbf{z} \mid do(\mathbf{x}))$ and $P(\mathbf{y} \mid do(\mathbf{z}))$ are themselves identifiable, and combine appropriately to form $P(\mathbf{y} \mid do(\mathbf{x}))$, we need two additional assumptions: there are no back-door paths from $\mathbf{X}$ to $\mathbf{Z}$, and all back-door paths from $\mathbf{Z}$ to $\mathbf{Y}$ are blocked by $\mathbf{X}$. It turns out that these three conditions imply that $P(\mathbf{y} \mid do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{y} \mid do(\mathbf{z}))P(\mathbf{z} \mid do(\mathbf{x}))$, and the latter two conditions further imply that the first term is identifiable by the back-door criterion and equal to $\sum_{\mathbf{z}} P(\mathbf{y} \mid \mathbf{z}, \mathbf{x})P(\mathbf{x})$, while the second term is equal to $P(\mathbf{z} \mid \mathbf{x})$. Whenever these three conditions hold, the effect of $\mathbf{X}$ on $\mathbf{Y}$ is identifiable. This is known as the *front-door criterion* (Pearl, 1995, 2000). The front-door criterion holds in the graph shown in Figure 2 (e).

Unfortunately, in some graphs neither the front-door, nor the back-door criterion holds. The simplest such graph, known as the bow arc graph due to its shape, is shown in Figure 1 (a). The back-door criterion fails since the confounder node is unobservable, while the front-door criterion fails since no intermediate variables between X and Y exist in the graph. While the failure of these two criteria does not imply non-identification, in fact the effect $P(y \mid do(x))$ is identifiable in Figure 2 (f), (g) despite this failure, a simple argument shows that $P(y \mid do(x))$ is not identifiable in the bow arc graph.

Theorem 10 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(x))$ in G shown in Figure 1 (a).*

Since we are interested in completely characterizing graphs where a given causal effect $P(\mathbf{y} \mid do(\mathbf{x}))$ is identifiable, it would be desirable to list difficult graphs like the bow arc graph which prevent identification of causal effects, in the hope of eventually making such a list complete and finding a way to identify effects in all graphs not on the

list. We start constructing this list by considering graphs which generalize the bow arc graph since they can contain more than two nodes, but which also inherit its difficult structure. We call such graphs C-trees.

Definition 11 (C-tree) *A graph G which is both a C-component and a tree is called a C-tree.*

We call a C-tree with a root node Y Y-rooted. The graphs in Figure 1 (a), (d), (e), (f), and (h) are Y-rooted C-trees. It turns out that in any Y-rooted C-tree, the effect of any subset of nodes, other than Y, on the root Y is not identifiable.

Theorem 12 *Let G be a Y-rooted C-tree. Let $\boldsymbol{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(\boldsymbol{x}))$.*

C-trees play a prominent role in the identification of *direct effects*. Intuitively, the direct effect of X on Y exists if there is an arrow from X to Y in the graph, and corresponds to the flow of influence along this arrow. However, simply considering changes in Y after fixing X is insufficient for isolating direct effect, since X can influence Y along other, longer front-door paths than the direct arrow. In order to disregard such influences, we also fix all other parents of Y (which as noted earlier removes all arrows incoming to these parents and thus to Y). The expression corresponding to the direct effect of X on Y is then $P(y \mid do(pa(y)))$. The following theorem links C-trees and direct effects.

Theorem 13 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*

This theorem might suggest that C-trees might play an equally strong role in identifying arbitrary effects on a single variable, not just direct effects. Unfortunately, this turns out not to be the case, due to the following lemma.

Lemma 14 (downward extension lemma) *Let $\boldsymbol{V}$ be the set of observable nodes in G, and $P(\boldsymbol{v})$ the observable distribution of models inducing G. Assume $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y} \mid do(\boldsymbol{x}))$. Let G' contain all the nodes and edges of G, and an additional node Z which is a child of all nodes in $\boldsymbol{Y}$. Then if $P(\boldsymbol{v}, z)$ is the observable distribution of models inducing G', then $P(\boldsymbol{v}, z), G' \not\vdash_{id} P(z \mid do(\boldsymbol{x}))$.*

Proof Let $|Z| = \prod_{Y_i \in \mathbf{Y}} |Y_i| = n$. By construction, $P(z \mid do(\mathbf{x})) = \sum_{\mathbf{y}} P(z \mid \mathbf{y}) P(\mathbf{y} \mid do(\mathbf{x}))$. Due to the way we set the arity of Z, $P(Z \mid \mathbf{Y})$ is an n by n matrix which acts as a linear map which transforms $P(\mathbf{y} \mid do(\mathbf{x}))$ into $P(z \mid do(\mathbf{x}))$. Since we can arrange this linear map to be one to one, any proof of non-identifiability of $P(\mathbf{y} \mid do(\mathbf{x}))$ immediately extends to the proof of non-identifiability of $P(z \mid do(\mathbf{x}))$. ■

What this lemma shows is that identification of effects on a singleton is not any simpler than the general problem of identification of effect on a set. To find difficult graphs which prevent identification of effects on sets, we consider a multi-root generalization of C-trees.

Definition 15 (c-forest) *A graph G which is both a C-component and a forest is called a C-forest.*

If a given C-forest has a set of root nodes $\mathbf{R}$, we call it $\mathbf{R}$-rooted. Graphs in Figure 3 (a), (b) are $\{Y1, Y2\}$-rooted C-forests. A naive way to generalize Theorem 12 would be to state that if G is an $\mathbf{R}$-rooted C-forest, then the effect of any set $\mathbf{X}$ that does not intersect $\mathbf{R}$ is not identifiable. However, as we later show, this is not true. Specifically, we later prove that $P(y1, y2 \mid do(x))$ in the graph in Figure 3 (a) is identifiable. To formulate the correct generalization of Theorem 12, we must understand what made C-trees difficult for the purposes of identifying effects on the root Y. It turned out that for particular function choices, the effects of ancestors of Y on Y precisely cancelled themselves out so even though Y itself was dependent on its parents, it was observationally indistinguishable from a constant function. To get the same canceling of effects with C-forests, we must define a more complex graphical structure.

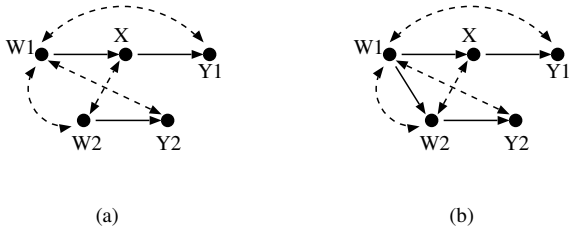


Figure 3: (a) A graph hedge-less for $P(y \mid do(x))$. (b) A graph containing a hedge for $P(y \mid do(x))$.

Definition 16 (hedge) *Let $\boldsymbol{X}, \boldsymbol{Y}$ be sets of variables in G. Let F, F' be $\boldsymbol{R}$-rooted C-forests in G such that F' is a subgraph of F, $\boldsymbol{X}$ only occur in F, and $\boldsymbol{R} \in An(\boldsymbol{Y})_{G_{\overline{x}}}$. Then F and F' form a hedge for $P(\boldsymbol{y} \mid do(\boldsymbol{x}))$.*

The graph in Figure 3 (b) contains a hedge for $P(y1, y2 \mid do(x))$. The mental picture for a hedge is as follows. We start with a C-forest F'. Then, F' grows new branches, while retaining the same root set, and becomes F. Finally, we "trim the hedge," by performing the action $do(\mathbf{x})$ which has the effect of removing some incoming arrows in $F \setminus F'$ (the subgraph of F consisting of vertices not a part of F'). Note that any Y-rooted C-tree and its root node Y form a hedge. The right generalization of Theorem 12 can be stated on hedges.

Theorem 17 *Let F, F' be subgraphs of G which form a hedge for $P(\boldsymbol{y} \mid do(\boldsymbol{x}))$. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y} \mid do(\boldsymbol{x}))$.*

Proof outline As before, assume binary variables. We let the causal mechanisms of one of the models consists entirely of bit parity functions. The second model also computes bit parity for every mechanism, except those nodes in F' which have parents in F ignore the values of those parents. It turns out that these two models are observationally indistinguishable. Furthermore, any intervention in $F \setminus F'$ will break the bit parity circuits of the models. This break will be felt at the root set $\mathbf{R}$ of the first model, but not of the second, by construction. ■

function **ID**(**y**, **x**, P, G)
INPUT: **x**,**y** value assignments, P a probability distribution, G a causal diagram.
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of P or **FAIL**(F,F').

1 if $\mathbf{x} = \emptyset$ return $\sum_{\mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.

2 if $\mathbf{V} \setminus An(\mathbf{Y})_G \neq \emptyset$
return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap An(\mathbf{Y})_G, \sum_{\mathbf{v} \setminus An(\mathbf{Y})_G} P, G_{An(\mathbf{Y})})$.

3 let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus An(\mathbf{Y})_{G_{\overline{\mathbf{x}}}}$.
if $\mathbf{W} \neq \emptyset$, return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.

4 if $C(G \setminus \mathbf{X}) = \{S_1, \ldots, S_k\}$
return $\sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_i \mathbf{ID}(s_i, \mathbf{v} \setminus s_i, P, G)$.

if $C(G \setminus \mathbf{X}) = \{S\}$

5 if $C(G) = \{G\}$, throw $\mathbf{FAIL}(G, G \cap S)$.

6 if $S \in C(G)$ return $\sum_{s \setminus \mathbf{y}} \prod_{\{i | V_i \in S\}} P(v_i \mid v_\pi^{(i-1)})$.

7 if $(\exists S')S \subset S' \in C(G)$ return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap S',$
$\prod_{\{i | V_i \in S'\}} P(V_i \mid V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S'), G_{S'})$.

Figure 4: A complete identification algorithm. **FAIL** propagates through recursive calls like an exception, and returns the hedge which witnesses non-identifiability. $V_\pi^{(i-1)}$ is the set of nodes preceding V_i in some topological ordering π in G.

Unlike the bow arc graph, and C-trees, hedges prevent identification of effects on multiple variables at once. Certainly a complete list of all possible difficult graphs must contain structures like hedges. But are there other kinds of structures that present problems? It turns out that the answer is "no," any time an effect is not identifiable in a causal model (if we make no restrictions on the type of function that can appear), there is a hedge structure involved. To prove that this is so, we need an algorithm which can identify any causal effect lacking a hedge. This algorithm, which we call **ID**, and which can be viewed as a simplified version of the identification algorithm due to Tian (2002), appears in Figure 4.

We will explain why each line of **ID** makes sense, and conclude by showing the operation of the algorithm on an example. The formal proof of soundness of **ID** can be found in the appendix. The first line merely asserts that if no action has been taken, the effect on **Y** is just the marginal of the observational distribution $P(\mathbf{v})$ on **Y**. The second line states that if we are interested in the effect on **Y**, it is sufficient to restrict our attention on the parts of the model ancestral to **Y**. One intuitive argument for this is that descendants of **Y** can be viewed as 'noisy versions' of **Y** and so any information they may impart which may be helpful for identification is already present in **Y**. On the other hand, variables which are neither ancestors nor descendants of **Y** lie outside the relevant causal chain entirely, and have no useful information to contribute.

Line 3 forces an action on any node where such an action would have no effect on **Y**—assuming we already acted on **X**. Since actions remove incoming arrows, we can view line 3 as simplifying the causal graph we consider by removing certain arcs from the graph, without affecting the overall answer. Line 4 is the key line of the algorithm, it decomposes the problem into a set of smaller problems using the key property of *c-component factorization* of causal models. If the entire graph is a single C-component already, further problem decomposition is impossible, and we must provide base cases. **ID** has three base cases. Line 5 fails because it finds two C-components, the graph G itself, and a subgraph S that does not contain any **X** nodes. But that is exactly one of the properties of C-forests that make up a hedge. In fact, it turns out that it is always possible to recover a hedge from these two c-components. Line 6 asserts that if there are no bidirected arcs from **X** to the other nodes in the current subproblem under consideration, then we can replace acting on **X** by conditioning, and thus solve the subproblem. Line 7 is the most complex case where **X** is partitioned into two sets, **W** which contain bidirected arcs into other nodes in the subproblem, and **Z** which do not. In this situation, identifying $P(\mathbf{y} \mid do(\mathbf{x}))$ from $P(\mathbf{v})$ is equivalent to identifying $P(\mathbf{y} \mid do(\mathbf{w}))$ from $P(\mathbf{V} \mid do(\mathbf{z}))$, since $P(\mathbf{y} \mid do(\mathbf{x})) = P(\mathbf{y} \mid do(\mathbf{w}), do(\mathbf{z}))$. But the term $P(\mathbf{V} \mid do(\mathbf{z}))$ is identifiable using the previous base case, so we can consider the subproblem of identifying $P(\mathbf{y} \mid do(\mathbf{w}))$.

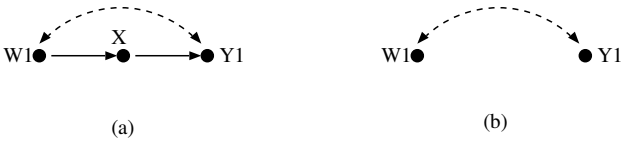


Figure 5: Subgraphs of G used for identifying $P_x(y_1, y_2)$.

We give an example of the operation of the algorithm by identifying $P_x(y_1, y_2)$ from $P(\mathbf{v})$ in the graph shown in in Figure 3 (a). Since $G = G_{An(\{Y_1,Y_2\})}$, $C(G \setminus \{X\}) = \{G\}$, and $\mathbf{W} = \{W_1\}$, we invoke line 3 and attempt to identify $P_{x,w}(y_1, y_2)$. Now $C(G \setminus \{X, W\}) = \{Y_1, W_2 \to Y_2\}$, so we invoke line 4. Thus the original problem reduces to identifying $\sum_{w_2} P_{x,w_1,w_2,y_2}(y_1) P_{w,x,y_1}(w_2, y_2)$. Solving for the second expression, we trigger line 2, noting that we can ignore nodes which are not ancestors of W_2 and Y_2, which means $P_{w,x,y_1}(w_2, y_2) = P(w_2, y_2)$. Solving for the first expression, we first trigger line 2 also, obtaining $P_{x,w_1,w_2,y_2}(y_1) = P_{x,w}(y_1)$. The corresponding G is shown in Figure 5 (a). Next, we trigger line 7, reducing the problem to computing $P_w(y_1)$ from $P(Y_1 \mid X, W_1)P(W_1)$. The corresponding G is shown in Figure 5 (b). Finally, we trigger line 2, obtaining $P_w(y_1) = \sum_{w_1} P(y_1 \mid x, w_1)P(w_1)$. Putting everything together, we obtain: $P_x(y_1, y_2) = \sum_{w_2} P(y_1, w_2) \sum_{w_1} P(y_1 \mid x, w_1)P(w_1)$.

As mentioned earlier, whenever the algorithm fails at line 5, it is possible to recover a hedge from the C-components S and G considered for the subproblem where the failure occurs. In fact, it can be shown that this hedge implies the non-identifiability of the original query with which the algorithm was invoked, which implies the following result.

Theorem 18 *ID is complete.*

The completeness of **ID** implies that hedges can be used to characterize all cases where effects of the form $P(\mathbf{y} \mid do(\mathbf{x}))$ cannot be identified from the observational distribution $P(\mathbf{v})$.

Theorem 19 (hedge criterion) *$P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y} \mid do(\boldsymbol{x}))$ if and only if G contains a hedge for some $P(\boldsymbol{y}' \mid do(\boldsymbol{x}'))$, where $\boldsymbol{y}' \subseteq \boldsymbol{y}$, $\boldsymbol{x}' \subseteq \boldsymbol{x}$.*

We close this section by considering identification of *conditional effects* of the form $P(\mathbf{y} \mid do(\mathbf{x}), \mathbf{z})$ which are defined to be equal to $P(\mathbf{y}, \mathbf{z} \mid do(\mathbf{x}))/P(\mathbf{z} \mid do(\mathbf{x}))$. Such expressions are a formalization of an intuitive notion of "effect of action in the presence of non-contradictory evidence," for instance the effect of smoking on lung cancer incidence rates in a particular age group (as opposed to the effect of smoking on cancer in the general population). We say that evidence $\mathbf{z}$ is non-contradictory since it is conceivable to consider questions where the evidence $\mathbf{z}$ stands in logical contradiction to the proposed hypothetical action $do(\mathbf{x})$: for instance what is the effect of smoking on cancer among the non-smokers. Such counterfactual questions will be considered in the next section. Conditioning can both help and hinder identifiability. $P(y \mid do(x))$ is not identifiable in the graph shown in Figure 6 (a), while it is identifiable in the graph shown in Figure 6 (b). Conditioning reverses the situation. In Figure 6 (a), conditioning on Z renders Y independent of any changes to X, making $P_x(y \mid z)$ equal to $P(y \mid z)$. On the other hand, in Figure 6 (b), conditioning on Z makes X and Y dependent, resulting in $P_x(y \mid z)$ becoming non-identifiable.

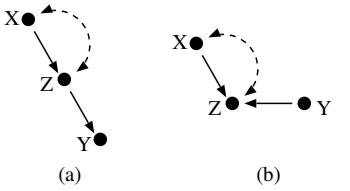


Figure 6: (a) Causal graph with an identifiable conditional effect $P(y \mid do(x), z)$. (b) Causal graph with a non-identifiable conditional effect $P(y \mid do(x), z)$.

We would like to reduce the problem of identifying conditional effects to the familiar problem of identifying causal effects without evidence for which we already have a complete algorithm. Fortunately, rule 2 of do-calculus provides us with a convenient way of converting the unwanted evidence $\mathbf{z}$ into actions $do(\mathbf{x})$ which we know how to handle. The following convenient lemma allows us to remove as many evidence variables as possible from a conditional effect.

Theorem 20 *For any G and any conditional effect $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ there exists a unique maximal set $\boldsymbol{Z} = \{Z \in \boldsymbol{W} \mid P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w}) = P_{\boldsymbol{x},z}(\boldsymbol{y} \mid \boldsymbol{w} \setminus \{z\})\}$ such that rule 2 applies to $\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$. In other words, $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y} \mid \boldsymbol{w} \setminus \boldsymbol{z})$.*

Of course Theorem 20 does not guarantee that the entire set $\mathbf{z}$ can be handled in this way. In many cases, even after rule 2 is applied, some set of evidence will remain in the expression. Fortunately, the following result implies that identification of unconditional causal effects is all we need.

Theorem 21 *Let $\boldsymbol{Z} \subseteq \boldsymbol{W}$ be the maximal set such that $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y} \mid \boldsymbol{w} \setminus \boldsymbol{z})$. Then $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ is identifiable in G if and only if $P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}, \boldsymbol{w} \setminus \boldsymbol{z})$ is identifiable in G.*

The previous two theorems suggest a simple addition to **ID**, which we call **IDC**, shown in Figure 7, which handles identification of conditional causal effects.

function **IDC**(**y**, **x**, **z**, P, G)
INPUT: **x**,**y**,**z** value assignments, P a probability distribution,
G a causal diagram (an I-map of P).
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{z})$ in terms of P or **FAIL**(F,F').

1 if $(\exists Z \in \mathbf{Z})(\mathbf{Y} \perp\!\!\!\perp Z \mid \mathbf{X}, \mathbf{Z} \setminus \{Z\})_{G_{\overline{\mathbf{X}},\underline{Z}}}$,
return **IDC**$(\mathbf{y}, \mathbf{x} \cup \{z\}, \mathbf{z} \setminus \{z\}, P, G)$.

2 else let $P' =$ **ID**$(\mathbf{y} \cup \mathbf{z}, \mathbf{x}, P, G)$.
return $P' / \sum_{\mathbf{y}} P'$.

Figure 7: A complete identification algorithm for conditional effects.

Theorem 22 ***IDC*** *is sound and complete.*

Proof This follows from Theorems 20 and 21. ■

We conclude this section by showing that our notion of a causal theory as a set of independencies embodied by the causal graph, together with rules of probability and do-calculus is complete for computing causal effects, if we also take statistical data embodied by $P(\mathbf{v})$ as axiomatic.

Theorem 23 *The rules of do-calculus are complete for identifying effects of the form $P(\boldsymbol{y} \mid do(\boldsymbol{x}), \boldsymbol{z})$, where $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ are arbitrary sets.*

Proof The proofs of soundness of **ID** and **IDC** in the appendix use do-calculus. This implies every line of the algorithms we presented can be rephrased as a sequence of do-calculus manipulations. But **ID** and **IDC** are also complete, which implies the conclusion. ■

4. Identification of Counterfactuals

While effects of actions have an intuitive interpretation as downward flow, the interpretation of counterfactuals, or what-if questions is more complex. An informal counterfactual statement in natural language such as "would I have a headache had I taken an aspirin" talks about multiple worlds: the actual world, and other, hypothetical worlds which differ in some small respect from the actual world (e.g., the aspirin was taken), while in most other respects are the same. In this paper, we represent the actual world by a causal model in its natural state, devoid of any interventions, while the alternative worlds are represented by submodels $M_{\mathbf{x}}$ where the action $do(\mathbf{x})$ implements the hypothetical change from the actual state of affairs considered. People make sense

of informal statements involving multiple, possibly conflicting worlds because they expect not only the causal rules to be invariant across these worlds (e.g., aspirin helps headaches in all worlds), but the worlds themselves to be similar enough where evidence in one world has ramifications in another. For instance, if we find ourselves with a headache, we expect the usual causes of our headache to also operate in the hypothetical world, interacting there with the preventative influence of aspirin. In our representation of counterfactuals, we model this interaction between worlds by assuming that the world histories or background contexts, represented by the unobserved **U** variables are shared across all hypothetical worlds.

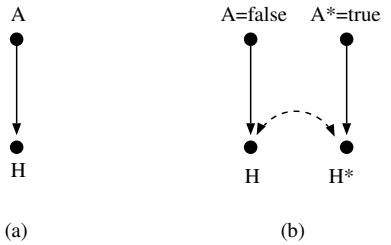


Figure 8: (a) A causal graph for the aspirin/headache domain (b) A corresponding twin network graph for the query $P(H^*_{a^*=true} \mid A = false)$.

We illustrate the representation method for counterfactuals we introduced in Section 2 by modeling our example question "would I have a headache had I taken an aspirin?" The actual world referenced by this query is represented by a causal model containing two variables, headache and aspirin, with aspirin being a parent of headache, see Figure 8 (a). In this world, we observe that aspirin has value false. The hypothetical world is represented by a submodel where the action $do(aspirin = true)$ has been taken. To distinguish nodes in this world we augment their names with an asterisk. The two worlds share the background variables **U**, and so can be represented by a single causal model with the graph shown in Figure 8 (b). Our query is represented by the distribution $P(H^*_{a^*=true} \mid A = false)$, where H is headache, and A is aspirin. Note that the nodes $A^* = true$ and $A = false$ in Figure 8 (b) do not share a bidirected arc. This is because an intervention $do(a^* = true)$ removes all incoming arrows to A^*, which removes the bidirected arc between A^* and A.

The graphs representing two hypothetical worlds invoked by a counterfactual query like the one shown in Figure 8 (b) are called *twin network graphs*, and were first proposed as a way to represent counterfactuals by Balke and Pearl (1994b) and Balke and Pearl (1994a). In addition, Balke and Pearl (1994b) proposed a method for evaluating expressions like $P(H^*_{a^*=true} \mid A = false)$ when all parameters of a causal model are known. This method can be explained as follows. If we forget the causal and counterfactual meaning behind the twin network graph, and simply view it as a Bayesian network, the query $P(H^*_{a^*=true} \mid A = false)$ can be evaluated using any of the standard inference algorithms available, provided we have access to all conditional probability tables generated by **F** and **U** of a causal model which gave rise to the twin network graph. In practice, however, complete knowledge of the model is too much to ask for; the functional relationships as well as the distribution $P(\mathbf{u})$ are not known

exactly, though some of their aspects can be inferred from the observable distribution $P(\mathbf{v})$.

Instead, the typical state of knowledge of a causal domain is the statistical behavior of the observable variables in the domain, summarized by the distribution $P(\mathbf{v})$, together with knowledge of causal directionality, obtained either from expert judgment (e.g., we know that visiting the doctor does not make us sick, though disease and doctor visits are highly correlated), or direct experimentation (e.g., it's easy to imagine an experiment which establishes that wet grass does not cause sprinklers to turn on). We already used these two sources of knowledge in the previous section as a basis for computing causal effects. Nevertheless, there are reasons to consider computing counterfactual quantities from experimental, rather than observational studies. In general, a counterfactual can posit worlds with features contradictory to what has actually been observed. For instance, questions resembling the headache/aspirin question we used as an example are actually frequently asked in epidemiology in the more general form where we are interested in estimating the effect of a treatment x on the outcome variable Y for the patients that were not treated (x'). In our notation, this is just our familiar expression $P(Y_x \mid X = x')$. The problem with questions such as these is that no experimental setup exists in which someone is both given and not given treatment. Therefore, it makes sense to ask under what circumstances we can evaluate such questions even if we are given as input every experiment that is possible to perform in principle on a given causal model. In our framework the set of all experiments is denoted as P_*, and is formally defined as $\{P_{\mathbf{x}} \mid \mathbf{x} \text{ is any set of values of } \mathbf{X} \subseteq \mathbf{V}\}$. The question that we ask in this section, then, is whether it is possible to identify a query $P(\gamma \mid \delta)$, where γ, δ are conjunctions of counterfactual events (with δ possibly empty), from the graph G and the set of all experiments P_*. We can pose the problem in this way without loss of generality since we already developed complete methods for identifying members of P_* from G and $P(\mathbf{v})$. This means that if for some reason using P_* as input is not realistic we can combine the methods which we will develop in this section with those in the previous section to obtain identification results for $P(\gamma \mid \delta)$ from G and $P(\mathbf{v})$.

Before tackling the problem of identifying counterfactual queries from experiments, we extend our example in Figure 8 (b) to a general graphical representation for worlds invoked by a counterfactual query. The twin network graph is a good first attempt at such a representation. It is essentially a causal diagram for a model encompassing two potential worlds. Nevertheless, the twin network graph suffers from a number of problems. Firstly, it can easily come to pass that a counterfactual query of interest would involve three or more worlds. For instance, we might be interested in how likely the patient would be to have a symptom Y given a certain dose x of drug X, assuming we know that the patient has taken dose x' of drug X, dose d of drug D, and we know how an intermediate symptom Z responds to treatment d. This would correspond to the query $P(y_x \mid x', z_d, d)$, which mentions three worlds, the original model M, and the submodels M_d, M_x. This problem is easy to tackle—we simply add more than two submodel graphs, and have them all share the same $\mathbf{U}$ nodes. This simple generalization of the twin network model was considered by Avin et al. (2005), and was called there the parallel worlds graph. Figure 9 shows the original causal graph and the parallel worlds graph for $\gamma = y_x \wedge x' \wedge z_d \wedge d$.

The other problematic feature of the twin network graph, which is inherited by the parallel worlds graph, is that multiple nodes can sometimes correspond to the same random variable. For example, in Figure 9 (b), the variables Z and Z_x are represented by distinct nodes, although it's easy to show that since Z is not a descendant of X,

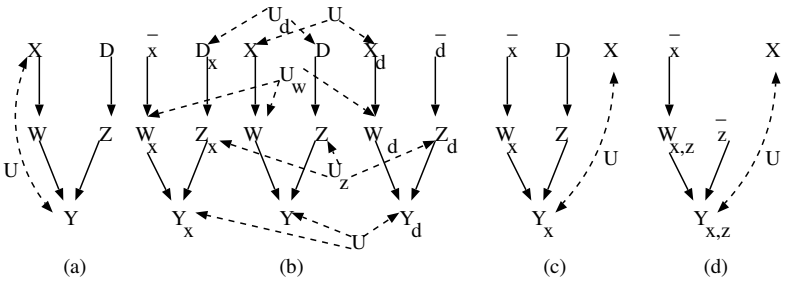


Figure 9: Nodes fixed by actions denoted with an overline, signifying that all incoming arrows are cut. (a) Original causal diagram (b) Parallel worlds graph for $P(y_x \mid x', z_d, d)$ (the two nodes denoted by U are the same). (c) Counterfactual graph for $P(y_x \mid x', z_d, d)$. (d) Counterfactual graph for $P(y_{x,z} \mid x')$.

$Z = Z_x$. These equality constraints among nodes can make the d-separation criterion misleading if not used carefully. For instance, $Y_x \perp\!\!\!\perp D_x \mid Z$ even though using d-separation in the parallel worlds graph suggests the opposite. This sort of problem is fairly common in causal models which are not *faithful* (Spirtes et al., 1993) or *stable* (Pearl, 2000), in other words in models where d-separation statements in a causal diagram imply independence in a distribution, but not vice versa. However, lack of faithfulness usually arises due to "numeric coincidences" in the observable distribution. In this case, the lack of faithfulness is "structural," in a sense that it is possible to refine parallel worlds graphs in such a way that the node duplication disappears, and the attendant independencies not captured by d-separation are captured by d-separation in refined graphs.

This refinement has two additional beneficial side effects. The first is that by removing node duplication, we also determine which syntactically distinct counterfactual variables correspond to the same random variable. By identifying such equivalence classes of counterfactual variables, we guarantee that syntactically different variables are in fact different, and this makes it simpler to reason about counterfactuals in order to identify them. For instance, a counterfactual $P(y_x, y')$ may either be non-identifiable or inconsistent (and so identifiable to equal 0), depending on whether Y_x and Y are the same variable. The second benefit of this refinement is that resulting graphs are generally much smaller and less cluttered than parallel worlds graphs, and so are easier to understand. Compare, for instance, the graphs in Figure 9 (b) and Figure 9 (c). To rid ourselves of duplicates, we need a formal way of determining when variables from different submodels are in fact the same. The following lemma does this.

Lemma 24 *Let M be a model inducing G containing variables α, β with the following properties:*

- *α and β have the same domain of values.*
- *There is a bijection f from $Pa(\alpha)$ to $Pa(\beta)$ such that a parent γ and $f(\gamma)$ have the same domain of values.*
- *The functional mechanisms of α and β are the same (except whenever the function for α uses the parent γ, the corresponding function for β uses $f(\gamma)$).*

Assume an observable variable set $\mathbf{Z}$ was observed to attain values $\mathbf{z}$ in $M_{\mathbf{x}}$, the submodel obtained from M by forcing another observable variable set $\mathbf{X}$ to attain values $\mathbf{x}$. Assume further that for each $\gamma \in Pa(\alpha)$, either $f(\gamma) = \gamma$, or γ and $f(\gamma)$ attain the same values (whether by observation or intervention). Then α and β are the same random variable in $M_{\mathbf{x}}$ with observations $\mathbf{z}$.

Proof This follows from the fact that variables in a causal model are functionally determined from their parents. ■

If two distinct nodes in a causal diagram represent the same random variable, the diagram contains redundant information, and the nodes must be merged. If two nodes, say corresponding to $Y_{\mathbf{x}}, Y_{\mathbf{z}}$, are established to be the same in G, they are merged into a single node which inherits all the children of the original two. These two nodes either share their parents (by induction) or their parents attain the same values. If a given parent is shared, it becomes the parent of the new node. Otherwise, we pick one of the parents arbitrarily to become the parent of the new node. This operation is summarized by the following lemma.

Lemma 25 *Let $M_{\mathbf{x}}$ be a submodel derived from M with set $\mathbf{Z}$ observed to attain values $\mathbf{z}$, such that Lemma 24 holds for α, β. Let M' be a causal model obtained from M by merging α, β into a new node ω, which inherits all parents and the functional mechanism of α. All children of α, β in M' become children of ω. Then $M_{\mathbf{x}}, M'_{\mathbf{x}}$ agree on any distribution consistent with $\mathbf{z}$ being observed.*

Proof This is a direct consequence of Lemma 24. ■

The new node ω we obtain from Lemma 25 can be thought of as a new counterfactual variable. As mentioned in section 2, such variables take the form $Y_{\mathbf{x}}$ where Y is the variable in the original causal model, and $\mathbf{x}$ is a subscript specifying the action which distinguishes the counterfactual. Since we only merge two variables derived from the same original, specifying Y is simple. But what about the subscript? Intuitively, the subscript of ω contains those fixed variables which are ancestors of ω in the graph G' of M'. Formally the subscript is $\mathbf{w}$, where $\mathbf{W} = An(\omega)_{G'} \cap \mathbf{sub}(\gamma)$, where the $\mathbf{sub}(\gamma)$ corresponds to those nodes in G' which correspond to subscripts in γ. Since we replaced α, β by ω, we replace any mention of α, β in our given counterfactual query $P(\gamma)$ by ω. Note that since α, β are the *same*, their value assignments must be the same (say equal to y). The new counterfactual ω inherits this assignment.

We summarize the inductive applications of Lemma 24, and 25 by the **make-cg** algorithm, which takes γ and G as arguments, and constructs a version of the parallel worlds graph without duplicate nodes. We call the resulting structure the *counterfactual graph* of γ, and denote it by G_γ. The algorithm is shown in Figure 10.

There are three additional subtleties in **make-cg**. The first is that if variables $Y_{\mathbf{x}}, Y_{\mathbf{z}}$ were judged to be the same by Lemma 24, but γ assigns them different values, this implies that the original set of counterfactual events γ is inconsistent, and so $P(\gamma) = 0$. The second is that if we are interested in identifiability of $P(\gamma)$, we can restrict ourselves to the ancestors of γ in G'. We can justify this using the same intuitive argument we used in Section 3 to justify Line 2 in **ID**. The formal proof for line 2 we provide in the appendix applies with little change to **make-cg**. Finally, because the algorithm can make an arbitrary choice picking a parent of ω each time Lemma 25 is applied, both

function **make-cg**(G, γ)
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: A counterfactual graph G_γ, and either a set of events γ' s.t. $P(\gamma') = P(\gamma)$ or **INCONSISTENT**

- Construct a submodel graph $G_{\mathbf{x}_i}$ for each action $do(\mathbf{x}_i)$ mentioned in γ. Construct the parallel worlds graph G' by having all such submodel graphs share their corresponding U nodes.
- Let π be a topological ordering of nodes in G', let $\gamma' := \gamma$.
- Apply Lemmas 24 and 25, in order π, to each observable node pair α, β derived from the same variable in G. For each α, β that are the same, do:
  - Let G' be modified as specified in Lemma 25.
  - Modify γ' by renaming all occurrences of β to α.
  - If $\mathbf{val}(\alpha) \neq \mathbf{val}(\beta)$, return G', **INCONSISTENT**.
- return $(G'_{An(\gamma')}, \gamma')$, where $An(\gamma')$ is the set of nodes in G' ancestral to nodes corresponding to variables mentioned in γ'.

Figure 10: An algorithm for constructing counterfactual graphs

the counterfactual graph G', and the corresponding modified counterfactual γ' are not unique. This does not present a problem, however, as any such graph is acceptable for our purposes.

We illustrate the operation of **make-cg** by showing how the graph in Figure 9 (c) is derived from the graph in Figure 9 (b). We start the application of Lemma 24 from the topmost observable nodes, and conclude that the node pairs D_x, D, and X_d, X have the same functional mechanisms, and the same parent set (in this case the parents are unobservable nodes U_d for the first pair, and U for the second). We then use Lemma 25 to obtain the graph shown in Figure 11 (a). Since the node pairs are the same, we pick the name of one of the nodes of the pair to serve as the name of the new node. In our case, we picked D and X. Note that for this graph, and all subsequent intermediate graphs we generate, we use the convention that if a merge creates a situation where an unobservable variable has a single child, that variable is omitted from the graph. For instance, in Figure 11 (a), the variable U_d, and its corresponding arrow to D omitted.

Next, we apply Lemma 24 for the node pair W_d, W. In this case, the functional mechanisms are once again the same, while the parents of W_d, W are X and U_w. We can also apply Lemma 24 twice to conclude that Z, Z_x and Z_d are in fact the same node, and so can be merged. The functional mechanisms of these three nodes are the same, and they share the parent U_z. As far as the parents of this triplet, the U_z parent is shared by all three, while Z, Z_x share the parent D, and Z_d has a separate parent d, fixed by intervention. However, in our counterfactual query, which is $P(y_x \mid x', z_d, d)$, the variable D happens to be observed to attain the value d, the same as the intervention value for the parent of Z_d. This implies that for the purposes of the Z, Z_x, Z_d triplet, their D-derived parents share the same value, which allows us to conclude they are the

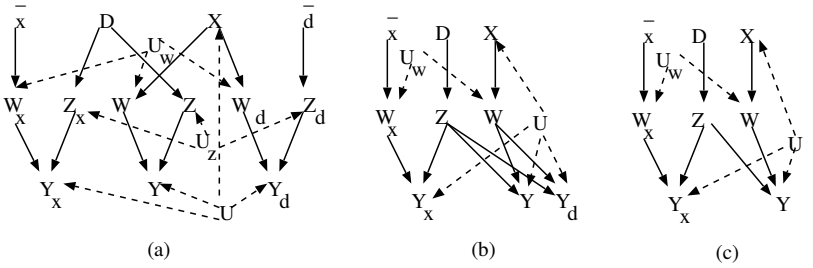


Figure 11: Intermediate graphs obtained by **make-cg** in constructing the counterfactual graph for $P(y_x \mid x', z_d, d)$ from Figure 9 (b).

same random variable. The intuition here is that while intervention and observation are not the same operation, they have the same effect if the relevant U variables happen to react in the same way to both the given intervention, and the given observation (this is the essence of the Axiom of Composition discussed by Pearl (2000).) In our case, U variables react the same way because the parallel worlds share all unobserved variables.

There is one additional subtlety in performing the merge of the triplet Z, Z_x, Z_d. If we examine our query $P(y_x \mid x', z_d, d)$, we notice that Z_d, or more precisely its value, appears in it. When we merge nodes, we only use one name out of the original two. It's possible that some of the old names appear in the query, which means we must replace all references to the old, pre-merge nodes with the new post-merge name we picked. Since we picked the name Z for the newly merged node, we replace the reference to Z_d in our query by the reference to Z, so our modified query is $P(y_x \mid x', z, d)$. Since the variables were established to be the same, this is a safe syntactic transformation.

After W_d, W, and the Z, Z_x, Z_d triplet are merged, we obtain the graph in Figure 11 (b). Finally, we apply Lemma 24 one more time to conclude Y and Y_d are the same variable, using the same reasoning as before. After performing this final merge, we obtain the graph in Figure 11 (c). It's easy to see that Lemma 24 no longer applies to any node pair: W and W_x differ in their X-derived parent, and Y, and Y_x differ on their W-derived parent, which was established inductively. The final operation which **make-cg** performs is restricting the graph in Figure 11 (b) to variables actually relevant for computing the (potentially syntactically modified) query it was given as input, namely $P(y_x \mid x', z, d)$. These relevant variables are ancestral to variables in the query in the final intermediate graph we obtained. In our case, we remove nodes W and Y (and their adjacent edges) from consideration, to finally obtain the graph in Figure 9 (c), which is a counterfactual graph for our query.

Having constructed a graphical representation of worlds mentioned in counterfactual queries, we can turn to identification. We construct two algorithms for this task, the first is called **ID*** and works for unconditional queries, while the second, **IDC***, works on queries with counterfactual evidence and calls the first as a subroutine. These are shown in Figure 12.

These algorithms make use of the following notation: **sub**$(.)$ returns the set of subscripts, **var**$(.)$ the set of variables, and **ev**$(.)$ the set of values (either set or observed) appearing in a given counterfactual conjunction (or set of counterfactual events), while

function **ID***(G, γ)
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: an expression for $P(\gamma)$ in terms of P_* or **FAIL**

1 if $\gamma = \emptyset$, return 1

2 if $(\exists x_{x'..} \in \gamma)$, return 0

3 if $(\exists x_{x..} \in \gamma)$, return **ID***$(G, \gamma \setminus \{x_{x..}\})$

4 $(G', \gamma') =$ **make-cg**(G, γ)

5 if $\gamma' =$ **INCONSISTENT**, return 0

6 if $C(G') = \{S^1, \ldots, S^k\}$,
return $\sum_{V(G') \setminus \gamma'} \prod_i$ **ID***$(G, s^i_{v(G') \setminus s^i})$

7 if $C(G') = \{S\}$ then,

8 if $(\exists \mathbf{x}, \mathbf{x}')$ s.t. $\mathbf{x} \neq \mathbf{x}', \mathbf{x} \in \mathbf{sub}(S), \mathbf{x}' \in \mathbf{ev}(S)$,
throw **FAIL**

9 else, let $\mathbf{x} = \bigcup \mathbf{sub}(S)$
return $P_{\mathbf{x}}(\mathbf{var}(S))$

function **IDC***(G, γ, δ)
INPUT: G a causal diagram, γ, δ conjunctions of counterfactual events
OUTPUT: an expression for $P(\gamma \mid \delta)$ in terms of P_*, **FAIL**, or **UNDEFINED**

1 if **ID***$(G, \delta) = 0$, return **UNDEFINED**

2 $(G', \gamma' \wedge \delta') =$ **make-cg**$(G, \gamma \wedge \delta)$

3 if $\gamma' \wedge \delta' =$ **INCONSISTENT**, return 0

4 if $(\exists y_{\mathbf{x}} \in \delta')$ s.t. $(Y_{\mathbf{x}} \perp\!\!\!\perp \gamma')_{G'_{\underline{y_{\mathbf{x}}}}}$,
return **IDC***$(G, \gamma'_{y_{\mathbf{x}}}, \delta' \setminus \{y_{\mathbf{x}}\})$

5 else, let $P' =$ **ID***$(G, \gamma' \wedge \delta')$. return $P'/P'(\delta)$

Figure 12: Counterfactual identification algorithms.

val$(.)$ is the value assigned to a given counterfactual variable. This notation is used to extract variables and values present in the original causal model from a counterfactual which refers to parallel worlds. As before, $C(G')$ is the set of maximal C-components of G', except we don't count nodes in G' fixed by interventions as part of any C-component. $V(G')$ is the set of observable nodes of G' not fixed by interventions. Following Pearl (2000), $G'_{\underline{y_{\mathbf{x}}}}$ is the graph obtained from G' by removing all outgoing arcs from $Y_{\mathbf{x}}$; $\gamma'_{y_{\mathbf{x}}}$ is obtained from γ' by replacing all descendant variables $W_{\mathbf{z}}$ of $Y_{\mathbf{x}}$

in γ' by $W_{\mathbf{z},y}$. A counterfactual $\mathbf{s_r}$, where $\mathbf{s}, \mathbf{r}$ are value assignments to sets of nodes, represents the event "the node set $\mathbf{S}$ attains values $\mathbf{s}$ under intervention $do(\mathbf{r})$." For instance, the term $s^i_{v(g')\setminus s^i}$ stands for the event "the node set S^i attains values s^i under the intervention $do(v(g') \setminus s^i)$," in other words under the intervention where we fix the values of all observable nodes in G' except those in S^i. Finally, we take $x_{x..}$ to mean some counterfactual variable derived from X where x appears in the subscript (the rest of the subscript can be arbitrary), which also attains value x.

The notation used in these algorithms is somewhat intricate, so we give an intuitive description of each line. We start with **ID***. The first line states that if γ is an empty conjunction, then its probability is 1, by convention. The second line states that if γ contains a counterfactual which violates the Axiom of Effectiveness (Pearl, 2000), then γ is inconsistent, and we return probability 0. The third line states that if a counterfactual contains its own value in the subscript, then it is a tautological event, and it can be removed from γ without affecting its probability. Line 4 invokes **make-cg** to construct a counterfactual graph G', and the corresponding relabeled counterfactual γ'. Line 5 returns probability 0 if an inconsistency was found during the construction of the counterfactual graph, for example, if two variables found to be the same in γ had different value assignments. Line 6 is analogous to Line 4 in the **ID** algorithm, it decomposes the problem into a set of subproblems, one for each C-component in the counterfactual graph. In the **ID** algorithm, the term corresponding to a given C-component S_i of the causal diagram was the effect of all variables not in S_i on variables in S_i, in other words $P_{\mathbf{v}\setminus s_i}(s_i)$, and the outermost summation on line 4 was over values of variables not in $\mathbf{Y}, \mathbf{X}$. Here, the term corresponding to a given C-component S^i of the counterfactual graph G' is the conjunction of counterfactual variables where each variable contains in its subscript all variables not in the C-component S^i, in other words $\mathbf{v}(G') \setminus s^i$, and the outermost summation is over observable variables not in γ', that is over $\mathbf{v}(G') \setminus \gamma'$, where we interpret γ' as a set of counterfactuals, rather than a conjunction. Line 7 is the base case, where our counterfactual graph has a single C-component. There are two cases, corresponding to line 8 and line 9. Line 8 says that if γ' contains a "conflict," that is an inconsistent value assignment where at least one value is in the subscript, then we fail. Line 9 says if there are no conflicts, then its safe to take the union of all subscripts in γ', and return the effect of the subscripts in γ' on the variables in γ'.

The **IDC***, like its counterpart **IDC**, is shorter. The first line fails if δ is inconsistent. **IDC** did not have an equivalent line, since we can assume $P(\mathbf{v})$ is positive. The problem with counterfactual distributions is there is no simple way to prevent non-positive distributions spanning multiple worlds from arising, even if the original $P(\mathbf{v})$ was positive—hence the explicit check. The second line constructs the counterfactual graph, except since **make-cg** can only take conjunctions, we provide it with a joint counterfactual $\gamma \wedge \delta$. Line 3 returns 0 if an inconsistency was detected. Line 4 of **IDC*** is the central line of the algorithm and is analogous to line 1 of **IDC**. In **IDC**, we moved a value assignment $Z = z$ from being observed to being fixed if there were no back-door paths from Z to the outcome variables $\mathbf{Y}$ given the context of the effect of $do(\mathbf{x})$. Here in **IDC***, we move a counterfactual value assignment $Y_{\mathbf{x}} = y$ from being observed (that is being a part of δ), to being fixed (that is appearing in every subscript of γ') if there are no back-door paths from $Y_{\mathbf{x}}$ to the counterfactual of interest γ'. Finally, line 5 of **IDC*** is the analogue of line 2 of **IDC**, we attempt to identify a joint counterfactual probability, and then obtain a conditional counterfactual probability from the result.

We illustrate the operation of these algorithms by considering the identification of a query $P(y_x \mid x', z_d, d)$ we mentioned earlier. Since $P(x', z_d, d)$ is not inconsistent, we proceed to construct the counterfactual graph on line 2. Suppose we produce the graph in Figure 9 (c), where the corresponding modified query is $P(y_x \mid x', z, d)$. Since $P(y_x, x', z, d)$ is not inconsistent we proceed to the next line, which moves z, d (with d being redundant due to graph structure) to the subscript of y_x, to obtain $P(y_{x,z} \mid x')$, and calls **IDC*** with this query recursively. Note that since the subscripts in one of the variables of our query changed, the counterfactual graph generated will change as well. In particular, the invocation of **make-cg** with the joint distribution from which $P(y_{x,z} \mid x')$ is derived, namely $P(y_{x,z}, x')$, will result in the graph shown in Figure 9 (d). Since X' has a back-door path to $Y_{x,z}$ in this graph, we can no longer call **IDC*** recursively, so we invoke **ID*** with the query $P(y_{x,z}, x')$.

The first interesting line in **ID*** is line 6, which first computes $P(y_{x,z}, w_{x,z}, x')$ by C-component factorization, and then computes $P(y_{x,z}, x')$ from $P(y_{x,z}, w_{x,z}, x')$ by marginalizing over $W_{x,z}$.[2] Since the counterfactual graph for this query (Figure 9 (d)) has two C-components, $\{Y_{x,z}, X\}$ and $\{W_{x,z}\}$, $P(y_{x,z}, w_{x,z}, x') = P(y_{x,z,w}, x'_w)P(w_{x,z})$, which can be simplified by removing redundant subscripts to $P(y_{z,w}, x')P(w_x)$. Line 6 then recursively calls **ID*** with $P(y_{x,z,w}, x')$ and $P(w_x)$, multiplies the results and marginalizes over W_x. The first recursive call reaches line 9 with $P(y_{z,w}, x')$, which is identifiable as $P_{z,w}(y, x')$ from P_*. The second term is trivially identifiable as $P_x(w)$, which means our query is identifiable as $P' = \sum_w P_{z,w}(y, x')P_x(w)$, and the conditional query is equal to $P'/P'(x')$.

The definitions of **ID***, and **IDC*** reveal their close similarity to algorithms **ID** and **IDC** in the previous section. The major differences lie in the failure and success base cases, and slightly different subscript notation. This is not a coincidence, since a counterfactual graph can be thought of as a causal graph for a particular large causal model which happens to have some distinct nodes share the same causal mechanisms. This means that all the theorems and definitions used in the previous sections for causal diagrams transfer over without change to counterfactual graphs. Using this fact, we will show that **ID***, and **IDC*** are sound and complete for identifying $P(\gamma)$, and $P(\gamma \mid \delta)$ respectively.

Theorem 26 (soundness) *If **ID*** succeeds, the expression it returns is equal to $P(\gamma)$ in a given causal graph. Furthermore, if **IDC*** does not output **FAIL**, the expression it returns is equal to $P(\gamma \mid \delta)$ in a given causal graph, if that expression is defined, and **UNDEFINED** otherwise.*

Proof outline The first line merely states that the probability of an empty conjunction is 1, which is true by convention. Lines 2 and 3 follow by the Axiom of Effectiveness (Galles and Pearl, 1998). The soundness of **make-cg** has already been established, which implies the soundness of line 4. Line 6 decomposes the problem using c-component factorization. The soundness proof for this decomposition, also used in the previous section, is in the appendix. Line 9 asserts that if a set of counterfactual events does not contain conflicting value assignments to any variable, obtained either by observation or intervention, then taking the union of all actions of the events results in a consistent action. The probability of the set of events can then be computed from a submodel

2. Note that since $W_{x,z}$ is a counterfactual variable derived from W, it shares its domain with W. Therefore it makes sense when marginalizing to operate over the values of W, denoted by w in the subscript of the summation.

where this consistent action has taken place. A full proof of this is in the appendix. ■

To show completeness, we follow the same strategy we used in the previous section. We catalogue all difficult counterfactual graphs which arise from queries which cannot be identified from P_*. We then show these graphs arise whenever **ID*** and **IDC*** fail. This, together with the soundness theorem we already proved, implies that these algorithms are complete.

The simplest difficult counterfactual graph arises from the query $P(y_x, y'_{x'})$ named "probability of necessity and sufficiency" by Pearl (2000). This graph, shown in Figure 8 (b) with variable relabeling, is called the "w-graph" due to its shape (Avin et al., 2005). This query is so named because if $P(y_x, y'_{x'})$ is high, this implies that if the variable X is forced to x, variable Y is likely to be y, while if X is forced to some other value, Y is likely to not be y. This means that the action $do(x)$ is likely a necessary and sufficient cause of Y assuming value y, up to noise. The w-graph starts our catalogue of bad graphs with good reason, as the following lemma shows.

Lemma 27 *Assume X is a parent of Y in G. Then $P_*, G \not\vdash_{id} P(y_x, y'_{x'}), P(y_x, y')$ for any value pair y, y'.*

Proof See Avin et al. (2005). ■

The intuitive explanation for this result is that $P(y_x, y'_{x'})$ is derived from the joint distribution over the counterfactual variables in the w-graph, while if we restrict ourselves to P_*, we only have access to marginal distributions—one marginal for each possible world. Because counterfactual variables Y_x and $Y_{x'}$ share an unobserved parent U, they are dependent, and their joint distribution cannot be decomposed into a product of marginals. This means that the information encoded in the marginals is insufficient to uniquely determine the joint we are interested in. This intuitive argument can be generalized to a counterfactual graph with more than two nodes, the so-called "zig-zag graphs" an example of which is shown in Figure 13 (b).

Lemma 28 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1, \ldots, W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1, \ldots, w^k, z_{x'}), P(y_x, w^1, \ldots, w^k, z)$ for any value assignments $y, w^1, \ldots, w^k, z$.*

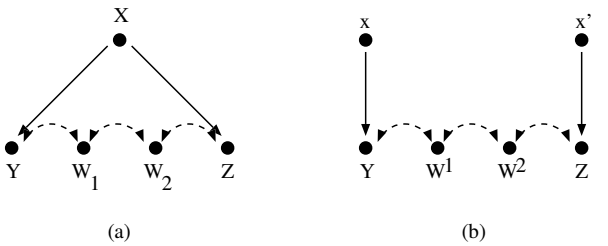


Figure 13: (a) Causal diagram (b) Corresponding counterfactual graph for the non-identifiable query $P(Y_x, W^1, W^2, Z_{x'})$.

The w-graph in Figure 8 (b) and the zig-zag graph in Figure 13 (b) have very special structure, so we don't expect our characterization to be complete with just these graphs. In order to continue, we must provide two lemmas which allow us to transform difficult graphs in various ways by adding nodes and edges, while retaining the non-identifiability of the underlying counterfactual from P_*.

Lemma 29 (downward extension lemma) *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let $\{y^1_{\boldsymbol{x}^1}, \ldots, y^n_{\boldsymbol{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, \ldots, Y^n$, and let P'_* be the set of all interventional distributions in models inducing G'. Let $\gamma' = (\gamma \setminus \{y^1_{\boldsymbol{x}^1}, \ldots, y^n_{\boldsymbol{x}^m}\}) \cup \{w_{\boldsymbol{x}^1}, \ldots, w_{\boldsymbol{x}^m}\}$, where w is an arbitrary value of W. Then $P'_*, G' \not\vdash_{id} P(\gamma')$.*

The first result states that non-identification on a set of parents (causes) translates into non-identification on children (effects). The intuitive explanation for this is that it is possible to construct a one-to-one function from the space of distributions on causes to the space of distributions on effects. If a given $P(\gamma)$ cannot be identified from P_*, this implies that there exist two models which agree on P_*, but disagree on $P(\gamma)$, where γ is a set of counterfactual causes. It is then possible to augment these models using the one-to-one function in question to obtain disagreement on $P(\delta)$, where δ is a set of counterfactual effects of γ. A more detailed argument is found in the appendix.

Lemma 30 (contraction lemma) *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If $(\exists w_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, $y \notin \boldsymbol{s}$, and $X \in An(W)_G$, then $Y \notin An(W)_G$.*
- *If $(\exists y_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, then $An(X)_G = \emptyset$.*
- *If $(Y_{\boldsymbol{w}}, X_{\boldsymbol{s}} \in \gamma)$, then $\boldsymbol{w}$ and $\boldsymbol{s}$ agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ and there's some isomorphism f assigning value pairs x, y to a value $f(x, y) = z$. Let γ' be obtained from γ as follows. For any $w_{\boldsymbol{s}} \in \gamma$:

- *If $W \notin \{X, Y\}$, and values x, y occur in $\boldsymbol{s}$, replace them by $f(x, y)$.*
- *If $W \notin \{X, Y\}$, and the value of one of X, Y occur in $\boldsymbol{s}$, replace it by some z consistent with the value of X or Y.*
- *If X, Y do not occur in γ, leave γ as is.*
- *If $W = Y$ and $x \in \boldsymbol{s}$, replace $w_{\boldsymbol{s}}$ by $f(x, y)_{\boldsymbol{s} \setminus \{x\}}$.*
- *otherwise, replace every variable pair of the form $Y_{\boldsymbol{r}} = y, X_{\boldsymbol{s}} = x$ by $Z_{\boldsymbol{r},\boldsymbol{s}} = f(x, y)$.*

Then $P_*, G' \not\vdash_{id} P(\gamma')$.

This lemma has a rather complicated statement, but the basic idea is very simple. If we have a causal model with a graph G where some counterfactual $P(\gamma)$ is not identifiable, then a coarser, more "near-sighted" view of G which merges two distinct variables with their own mechanisms into a single variable with a single mechanism

will not render $P(\gamma)$ identifiable. This is because merging nodes in the graph does not alter the model, but only our state of knowledge of the model. Therefore, whatever model pair was used to prove $P(\gamma)$ non-identifiable will remain the same in the new, coarser graph. The complicated statement of the lemma is due to the fact that we cannot allow arbitrary node merges, we must satisfy certain coherence conditions. For instance, the merge cannot create directed cycles in the graph.

It turns out that whenever **ID*** fails on $P(\gamma)$, the corresponding counterfactual graph contains a subgraph which can be obtained by a set of applications of the previous two lemmas to the w-graph and the zig-zag graphs. This allows an argument that shows $P(\gamma)$ cannot be identified from P_*.

Theorem 31 (completeness) *If **ID*** or **IDC*** fail, then the corresponding query is not identifiable from P_*.*

Since **ID*** is complete for $P(\gamma)$ queries, we can give a graphical characterization of counterfactual graphs where $P(\gamma)$ cannot be identified from P_*.

Theorem 32 *Let G_γ, γ' be obtained from **make-cg**(G, γ). Then $P_*, G \not\vdash_{id} P(\gamma)$ if and only if there exists a C-component $S \subseteq An(\gamma')_{G_\gamma}$ where some $X \in Pa(S)$ is set to x while at the same time either X is also a parent of another node in S and is set to another value x', or S contains a variable derived from X which is observed to be x'.*

Proof This follows from Theorem 31 and the construction of **ID***. ■

5. Conclusions

This paper considers a hierarchy of queries about relationships among variables in graphical causal models: associational relationships which can be obtained from observational studies, cause-effect relationships obtained by experimental studies, and counterfactuals, which are derived from parallel worlds resulting from hypothetical actions, possibly conflicting with available evidence. We consider the identification problem for this hierarchy, the task of computing a query from the given causal diagram and available information lower in the hierarchy.

We provide sound and complete algorithms for this identification problem, and a graphical characterization of non-identifiable queries where these algorithms must fail. Specifically, we provide complete algorithms for identifying causal effects and conditional causal effects from observational studies, and show that a graphical structure called a *hedge* completely characterizes all cases where causal effects are non-identifiable. As a corollary, we show that the three rules of do-calculus are complete for identifying effects. We also provide complete algorithms for identifying counterfactual queries (possibly conditional) from experimental studies. If we view the structure of the causal graph as experimentally testable, as is often the case in practice, this result can be viewed as giving a full characterization of testable counterfactuals assuming structural semantics.

These results settle important questions in causal inference, and pave the way for computing more intricate causal queries which involve nested counterfactuals, such as those defining direct and indirect effects (Pearl, 2001), and path-specific effects (Avin et al., 2005). The characterization of non-identifiable queries we provide defines

precisely the situations when such queries cannot be computed precisely, and must instead by approximated using methods such as bounding (Balke and Pearl, 1994a), instrumental variables (Pearl, 2000), or additional assumptions, such as linearity, which can make identification simpler.

Acknowledgments

The authors would like to thank Eleazar Eskin and Eun Yong Kang for discussing earlier versions of this work. This work was supported in part by AFOSR grant #F49620-01-1-0055, NSF grant #IIS-0535223, MURI grant #N00014-00-1-0617, and NLM grant #T15 LM07356.

Appendix A.

Here, we augment the intuitive proof outlines we gave in the main body of the paper with more formal arguments. We start with a set of results which were used to classify graphs with non-identifiable effects. In the proofs presented here, we will construct the distributions which make up our set of premises to be positive. This is because non-positive distributions present a number of technical difficulties, for instance d-separation and independence are not related in a straightforward way in such distributions, and conditional distributions may not be defined. We should mention, however, that distributions which span multiple hypothetical worlds which we discussed in Section 4 may be non-positive by definition.

Theorem 5 *If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map. Furthermore, the causal diagram induced by any semi-Markovian PCM M is a semi-Markovian I-map of the distribution $P(\boldsymbol{v}, \boldsymbol{u})$ induced by M.*

Proof It is not difficult to see that if we restrict d-separation queries to a subset of variables $\mathbf{W}$ in some graph G, the corresponding independencies in $P(\mathbf{w})$ will only hold whenever the d-separation statements hold. Furthermore, if we replace G by a latent projection L (Pearl, 2000), where we view variables $\mathbf{V} \setminus \mathbf{W}$ as hidden, independencies in $P(\mathbf{w})$ will only hold whenever the corresponding d-separation statement (extended to include bidirected arcs) holds in L. ■

Theorem 10 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(x))$ in G shown in Figure 1 (a).*

Proof We construct two causal models M^1 and M^2 such that $P^1(X, Y) = P^2(X, Y)$, and $P^1_x(Y) \neq P^2_x(Y)$. The two models agree on the following: all 3 variables are boolean, U is a fair coin, and $f_X(u) = u$. Let $\oplus$ denote the exclusive or (XOR) function. Then the value of Y is determined by the function $u \oplus x$ in M^1, while Y is set to 0 in M^2. Then $P^1(Y = 0) = P^2(Y = 0) = 1$, $P^1(X = 0) = P^2(X = 0) = 0.5$. Therefore, $P^1(X, Y) = P^2(X, Y)$, while $P^2_x(Y = 0) = 1 \neq P^1_x(Y = 0) = 0.5$. Note that while P is non-positive, it is straightforward to modify the proof for the positive case by letting f_Y functions in both models return 1 half the time, and the values outlined above half the time. ■

Theorem 12 *Let G be a Y-rooted C-tree. Let $\boldsymbol{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(\boldsymbol{x}))$.*

Proof We generalize the proof for the bow arc graph. We can assume without loss of generality that each unobservable U in G has exactly two observable children. We construct two models with binary nodes. In the first model, the value of all observable nodes is set to the bit parity (sum modulo 2) of the parent values. In the second model, the same is true for all nodes except Y, with the latter being set to 0 explicitly. All $\mathbf{U}$ nodes in both models are fair coins. Since G is a tree, and since every $U \in \mathbf{U}$ has exactly two children in G, every $U \in \mathbf{U}$ has exactly two distinct downward paths to Y in G. It's then easy to establish that Y counts the bit parity of every node in $\mathbf{U}$ twice in the first model. But this implies $P^1(Y = 1) = 0$.

Because bidirected arcs form a spanning tree over observable nodes in G, for any set of nodes $\mathbf{X}$ such that $Y \notin \mathbf{X}$, there exists $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_G$ and one child in $G \setminus An(\mathbf{X})_G$. Thus $P^1_{\mathbf{x}}(Y = 1) > 0$, but $P^2_{\mathbf{x}}(Y = 1) = 0$. It is straightforward to generalize this proof for the positive $P(\mathbf{v})$ in the same way as in Theorem 10. ■

Theorem 13 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y \mid do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*

Proof From Tian (2002), we know that whenever there is no subgraph G' of G, such that all nodes in G' are ancestors of Y, and G' is a C-component, $P_{pa(Y)}(Y)$ is identifiable. From Theorem 12, we know that if there is a Y-rooted C-tree containing a non-empty subset S of parents of Y, then $P_s(Y)$ is not identifiable. But it is always possible to extend the counterexamples which prove non-identification of $P_s(Y)$ with additional variables which are independent. ■

Theorem 17 *Let F, F' be subgraphs of G which form a hedge for $P(\boldsymbol{y} \mid do(\boldsymbol{x}))$. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y} \mid do(\boldsymbol{x}))$.*

Proof We first show $P_{\mathbf{x}}(\mathbf{r})$ is not identifiable in F. As before, we assume each U has two observable children. We construct two models with binary nodes. In M^1 every variable in F is equal to the bit parity of its parents. In M^2 the same is true, except all nodes in F' disregard the parent values in $F \setminus F'$. All $\mathbf{U}$ are fair coins in both models.

As was the case with C-trees, for any C-forest F, every $U \in \mathbf{U} \cap F$ has exactly two downward paths to $\mathbf{R}$. It is now easy to establish that in M^1, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^1$ twice, while in M^2, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^2 \cap F'$ twice. Thus, in both models with no interventions, the bit parity of $\mathbf{R}$ is even.

Next, fix two distinct instantiations of $\mathbf{U}$ that differ by values of $\mathbf{U}^*$. Consider the topmost node $W \in F$ with an odd number of parents in $\mathbf{U}^*$ (which exists because bidirected edges in F form a spanning tree). Then flipping the values of $\mathbf{U}^*$ once will flip the value W once. Thus the function from $\mathbf{U}$ to $\mathbf{V}$ induced by a C-forest F in M^1 and M^2 is one to one.

The above results, coupled with the fact that in a C-forest, $|\mathbf{U}| + 1 = |\mathbf{V}|$ implies that any assignment where $\sum \mathbf{r} \pmod 2 = 0$ is equally likely, and all other node assignments are impossible in both F and F'. Since the two models agree on all functions and

distributions in $F \setminus F'$, $\sum_{f'} P^1 = \sum_{f'} P^2$. It follows that the observational distributions are the same in both models.

As before, we can find $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_F$, and one child in $F \setminus An(\mathbf{X})_F$, which implies the probability of odd bit parity of $\mathbf{R}$ is 0.5 in M^1, and 0 in M^2.

Next, we note that the construction so far results in a non-positive distribution P. To rid our proof of non-positivity, we "soften" our two models with new unobservable binary U_R for every $R \in \mathbf{R}$ which assumes value 1 with very small probability p. Whenever U_R is 1, the node R flips its value, otherwise it keeps the value as defined above. Note that $P(\mathbf{v})$ will remain the same in both models because our augmentation is the same, and the previous unsoftened models agreed on $P(\mathbf{v})$. It's easy to see that the bit parity of R in both models will be odd only when an odd number of U_R assume values of 1. Because p is arbitrarily small, the probability of an odd parity is far smaller than the probability of even parity. Now consider what happens after $do(\mathbf{x})$. In M^2, the probability of odd bit parity stays the same. In M^1 before the addition of U_R, the probability was 0.5. But it's easy to see that U_R nodes change the bit parity of $\mathbf{R}$ in a completely symmetric way, so the probability of even parity remains 0.5.

This implies $P_{\mathbf{x}}(\mathbf{r})$ is not identifiable. Finally, to see that $P_{\mathbf{x}}(\mathbf{y})$ is not identifiable, augment our counterexample by nodes in $\mathbf{I} = An(\mathbf{Y}) \cap De(\mathbf{R})$. Without loss of generality, assume every node in $\mathbf{I}$ has at most one child. Let each node I in $\mathbf{I}$ be equal to the bit parity of its parents. Moreover, each I has an exogenous parent U_I independent of the rest of $\mathbf{U}$ which, with small probability p causes it to flip it's value. Then the bit parity of $\mathbf{Y}$ is even if and only if an odd number of $\mathbf{U_I}$ turn on. Moreover, it's easy to see $P(\mathbf{I} \mid \mathbf{R})$ is positive by construction. We can now repeat the previous argument. ■

Next, we provide the proof of soundness of **ID** and **IDC** using do-calculus. This both simplifies the proofs and allows us to infer do-calculus is complete from completeness of our algorithms. We will invoke do-calculus rules by just using their number, for instance "by rule 2." First, we prove that a joint distribution in a causal model can be represented as a product of interventional distributions corresponding to the set of c-component in the graph induced by the model.

Lemma 33 (c-component factorization) *Let M be a causal model with graph G. Let $\boldsymbol{y}, \boldsymbol{x}$ be value assignments. Let $C(G \setminus \boldsymbol{X}) = \{S_1, \ldots, S_k\}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{\boldsymbol{v} \setminus (\boldsymbol{y} \cup \boldsymbol{x})} \prod_i P_{\boldsymbol{v} \setminus s_i}(s_i)$.*

Proof A proof of this was derived by Tian (2002). Nevertheless, we reprove this result using do-calculus to help with our subsequent completeness results. Assume $\mathbf{X} = \emptyset$, $\mathbf{Y} = \mathbf{V}$, $C(G) = \{S_1, \ldots, S_k\}$, and let $A_i = An(S_i)_G \setminus S_i$. Then

$$\prod_i P_{\mathbf{v} \setminus s_i}(s_i) = \prod_i P_{a_i}(s_i) = \prod_i \prod_{V_j \in S_i} P_{a_i}(v_j \mid v_\pi^{(j-1)} \setminus a_i)$$

$$= \prod_i \prod_{V_j \in S_i} P(v_j \mid v_\pi^{(j-1)}) = \prod_i P(v_i \mid v_\pi^{(i-1)}) = P(\mathbf{v}).$$

The first identity is by rule 3, the second is by chain rule of probability. To prove the third identity, we consider two cases. If $A \in A_i \setminus V_\pi^{(j-1)}$, we can eliminate the intervention on A from the expression $P_{a_i}(v_j \mid v_\pi^{(j-1)})$ by rule 3, since $(V_j \perp\!\!\!\perp A \mid V_\pi^{(j-1)})_{G_{\overline{a_i}}}$.

If $A \in A_i \cap V_\pi^{(j-1)}$, consider any back-door path from A_i to V_j. Any such path with a node not in $V_\pi^{(j-1)}$ will be d-separated because, due to recursiveness, it must contain a blocked collider. Further, this path must contain bidirected arcs only, since all nodes on this path are conditioned or fixed. Because $A_i \cap S_i = \emptyset$, all such paths are d-separated. The identity now follows from rule 2. The last two identities are just grouping of terms, and application of chain rule.

Having proven that c-component factorization holds for $P(\mathbf{v})$, we want to extend the result to $P_{\mathbf{x}}(\mathbf{y})$. First, let's consider $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. This is just the distribution of the submodel $M_{\mathbf{x}}$. But $M_{\mathbf{x}}$ is just an ordinary causal model inducing $G \setminus \mathbf{X}$, so we can apply the same reasoning to obtain $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x}) = \prod_i P_{\mathbf{v} \setminus s_i}(s_i)$, where $C(G \setminus \mathbf{X}) = \{S_1, \ldots, S_k\}$. As a last step, it's easy to verify that $P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{v} \setminus (\mathbf{x} \cup \mathbf{y})} P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. ■

Lemma 34 *Let $\boldsymbol{X}' = \boldsymbol{X} \cap An(\boldsymbol{Y})_G$. Then $P_{\boldsymbol{x}}(\boldsymbol{y})$ obtained from P in G is equal to $P'_{\boldsymbol{x}'}(\boldsymbol{y})$ obtained from $P' = P(An(\boldsymbol{Y}))$ in $An(\boldsymbol{Y})_G$.*

Proof Let $\mathbf{W} = \mathbf{V} \setminus An(\mathbf{Y})_G$. Then the submodel $M_{\mathbf{w}}$ induces the graph $G \setminus \mathbf{W} = An(\mathbf{Y})_G$, and its distribution is $P' = P_{\mathbf{w}}(An(\mathbf{Y})) = P(An(\mathbf{Y}))$ by rule 3. Now $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}'}(\mathbf{y}) = P_{\mathbf{x}',\mathbf{w}}(\mathbf{y}) = P'_{\mathbf{x}'}(\mathbf{y})$ by rule 3. ■

Lemma 35 *Let $\boldsymbol{W} = (\boldsymbol{V} \setminus \boldsymbol{X}) \setminus An(\boldsymbol{Y})_{G_{\overline{\boldsymbol{x}}}}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}) = P_{\boldsymbol{x},\boldsymbol{w}}(\boldsymbol{y})$, where $\boldsymbol{w}$ are arbitrary values of $\boldsymbol{W}$.*

Proof Note that by assumption, $\mathbf{Y} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$ in $G_{\overline{\mathbf{x}},\overline{\mathbf{w}}}$. The conclusion follows by rule 3. ■

Lemma 36 *When the conditions of line 6 are satisfied, $P_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{s \setminus \boldsymbol{y}} \prod_{V_i \in S} P(v_i \mid v_\pi^{(i-1)})$.*

Proof If line 6 preconditions are met, then G local to that recursive call is partitioned into S and $\mathbf{X}$, and there are no bidirected arcs from $\mathbf{X}$ to S. The conclusion now follows from the proof of Lemma 33. ■

Lemma 37 *Whenever the conditions of the last recursive call of **ID** are satisfied, $P_{\boldsymbol{x}}$ obtained from P in the graph G is equal to $P'_{\boldsymbol{x} \cap S'}$ obtained from $P' = \prod_{V_i \in S'} P(V_i \mid V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S')$ in the graph S'.*

Proof It is easy to see that when the last recursive call executes, $\mathbf{X}$ and S partition G, and $\mathbf{X} \subset An(S)_G$. This implies that the submodel $M_{\mathbf{x} \setminus S'}$ induces the graph $G \setminus (\mathbf{X} \setminus S') = S'$. The distribution $P_{\mathbf{x} \setminus S'}$ of $M_{\mathbf{x} \setminus S'}$ is equal to P' by the proof of Lemma 33. It now follows that $P_{\mathbf{x}} = P_{\mathbf{x} \cap S', \mathbf{x} \setminus S'} = P'_{\mathbf{x} \cap S'}$. ■

Theorem 38 (soundness) *Whenever **ID** returns an expression for $P_{\boldsymbol{x}}(\boldsymbol{y})$, it is correct.*

Proof If $\mathbf{x} = \emptyset$, the desired effect can be obtained from P by marginalization, thus this base case is clearly correct. The soundness of all other lines except the failing line 5 has already been established. ■

Having established soundness, we show that whenever **ID** fails, we can recover a hedge for an effect involving a subset of variables involved in the original effect expression $P(\mathbf{y} \mid do(\mathbf{x}))$. This in turn implies completeness.

Theorem 39 *Assume **ID** fails to identify $P_{\boldsymbol{x}}(\boldsymbol{y})$ (executes line 5). Then there exist $\boldsymbol{X}' \subseteq \boldsymbol{X}$, $\boldsymbol{Y}' \subseteq \boldsymbol{Y}$ such that the graph pair G, S returned by the fail condition of **ID** contain as edge subgraphs C-forests F, F' that form a hedge for $P_{\boldsymbol{x}'}(\boldsymbol{y}')$.*

Proof Consider line 5, and G and $\mathbf{y}$ local to that recursive call. Let $\mathbf{R}$ be the root set of G. Since G is a single C-component, it is possible to remove a set of directed arrows from G while preserving the root set $\mathbf{R}$ such that the resulting graph F is an $\mathbf{R}$-rooted C-forest.

Moreover, since $F' = F \cap S$ is closed under descendants, and since only single directed arrows were removed from S to obtain F', F' is also a C-forest. $F' \cap \mathbf{X} = \emptyset$, and $F \cap \mathbf{X} \neq \emptyset$ by construction. $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{x}}}}$ by lines 2 and 3 of the algorithm. It's also clear that $\mathbf{y}, \mathbf{x}$ local to the recursive call in question are subsets of the original input. ■

Theorem 18 ***ID** is complete.*

Proof By the previous theorem, if **ID** fails, then $P_{\mathbf{x}'}(\mathbf{y}')$ is not identifiable in a subgraph $H = G_{An(\mathbf{Y}) \cap De(F)}$ of G. Moreover, $\mathbf{X} \cap H = \mathbf{X}'$, by construction of H. As such, it is easy to extend the counterexamples in Theorem 39 with variables independent of H, with the resulting models inducing G, and witnessing the non-identifiability of $P_{\mathbf{x}}(\mathbf{y})$. ■

Next, we prove the results necessary to establish completeness of **IDC**.

Lemma 40 *If rule 2 of do-calculus applies to a set $\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ then there are no d-connected paths to $\boldsymbol{Y}$ that pass through $\boldsymbol{Z}$ in neither $G_1 = G \setminus \boldsymbol{X}$ given $\boldsymbol{Z}, \boldsymbol{W}$ nor in $G_2 = G \setminus (\boldsymbol{X} \cup \boldsymbol{Z})$ given $\boldsymbol{W}$.*

Proof Clearly, there are no d-connected paths through $\mathbf{Z}$ in G_2 given $\mathbf{W}$. Consider a d-connected path through $Z \in \mathbf{Z}$ to $\mathbf{Y}$ in G_1, given $\mathbf{Z}, \mathbf{W}$. Note that this path must either form a collider at Z or a collider which is an ancestor of Z. But this must mean there is a back-door path from $\mathbf{Z}$ to $\mathbf{Y}$, which is impossible, since rule 2 is applicable to $\mathbf{Z}$ in G for $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$. Contradiction. ■

Theorem 20 *For any G and any conditional effect $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ there exists a unique maximal set $\boldsymbol{Z} = \{Z \in \boldsymbol{W} \mid P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w}) = P_{\boldsymbol{x},z}(\boldsymbol{y} \mid \boldsymbol{w} \setminus \{z\})\}$ such that rule 2 applies to $\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$. In other words, $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w}) = P_{\boldsymbol{x},z}(\boldsymbol{y} \mid \boldsymbol{w} \setminus \boldsymbol{z})$.*

Proof Fix two maximal sets $\mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{W}$ such that rule 2 applies to $\mathbf{Z}_1, \mathbf{Z}_2$ in G for $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$. If $\mathbf{Z}_1 \neq \mathbf{Z}_2$, fix $Z \in \mathbf{Z}_1 \setminus \mathbf{Z}_2$. By Lemma 40, rule 2 applies for $\{Z\} \cup \mathbf{Z}_2$ in G for $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$, contradicting our assumption.

Thus if we fix G and $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$, any set to which rule 2 applies must be a subset of the unique maximal set $\mathbf{Z}$. It follows that $\mathbf{Z} = \{Z \in \mathbf{W} \mid P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w}) = P_{\mathbf{x},z}(\mathbf{y} \mid \mathbf{w} \setminus \{z\})\}$. ■

Lemma 41 *Let F, F' form a hedge for $P_{\boldsymbol{x}}(\boldsymbol{y})$. Then $F \subseteq F' \cup \mathbf{X}$.*

Proof It has been shown that **ID** fails on $P_{\mathbf{x}}(\mathbf{y})$ in G and returns a hedge if and only if $P_{\mathbf{x}}(\mathbf{y})$ is not identifiable in G. In particular, edge subgraphs of the graphs G and S returned by line 5 of **ID** form the C-forests of the hedge in question. It is easy to check that a subset of $\mathbf{X}$ and S partition G. ■

We rephrase the statement of Theorem 21 somewhat, to reduce "algebraic clutter."

Theorem 21 *Let $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ be such that every $W \in \boldsymbol{W}$ has a back-door path to $\boldsymbol{Y}$ in $G \setminus \boldsymbol{X}$ given $\boldsymbol{W} \setminus \{W\}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y} \mid \boldsymbol{w})$ is identifiable in G if and only if $P_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{w})$ is identifiable in G.*

Proof If $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ is identifiable in G, then we can certainly identify $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$ by marginalization and division. The difficult part is to prove that if $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ is not identifiable then neither is $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$.

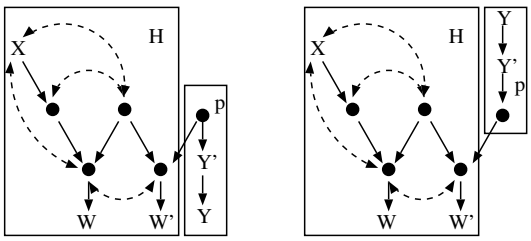


Figure 14: Inductive cases for proving non-identifiability of $P_x(y \mid w, w')$.

Assume $P_{\mathbf{x}}(\mathbf{w})$ is identifiable. Then if $P_{\mathbf{x}}(\mathbf{y} \mid \mathbf{w})$ were identifiable, we would be able to compute $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ by the chain rule. Thus our conclusion follows.

Assume $P_{\mathbf{x}}(\mathbf{w})$ is not identifiable. We also know that every $W \in \mathbf{W}$ contains a back-door path to some $Y \in \mathbf{Y}$ in $G \setminus \mathbf{X}$ given $\mathbf{W} \setminus \{W\}$. Fix such W and Y, along with a subgraph p of G which forms the witnessing back-door path. Consider also the hedge F, F' which witnesses the non-identifiability of $P_{\mathbf{x}'}(\mathbf{w}')$, where $\mathbf{X}' \subseteq \mathbf{X}, \mathbf{W}' \subseteq \mathbf{W}$.

Let $H = G_{De(F) \cup An(\mathbf{W}')_{G_{\overline{\mathbf{X}'}}}}$. We will attempt to show that $P_{\mathbf{x}'}(Y \mid \mathbf{w})$ is not identifiable in $H \cup p$. Without loss of generality, we make the following three assumptions. First, we restrict our attention to $\mathbf{W}'' \subseteq \mathbf{W}$ that occurs in $H \cup p$. Second, we assume p is a path segment which starts at H and ends at Y, and does not intersect H. Third, we assume all observable nodes in H have at most one child.

Consider the models M^1, M^2 from the proof of Theorem 17 which induce H. We extend the models by adding to them binary variables in p. Each variable $X \in p$ is equal to the bit parity of its parents, if it has any. If not, X behaves as a fair coin. If $Y \in H$ has a parent $X \in p$, the value of X is added to the bit parity computation Y makes.

Call the resulting models M^1_*, M^2_*. Because M^1, M^2 agreed on $P(H)$, and variables and functions in p are the same in both models, $P^1_* = P^2_*$. We will assume $\mathbf{w}''$ assigns 0 to every variable in $\mathbf{W}''$. What remains to be shown is that $P^1_{*\mathbf{x}}(y \mid \mathbf{w}'') \neq P^2_{*\mathbf{x}}(y \mid \mathbf{w}'')$. We will prove this by induction on the path structure of p. We handle the inductive cases first. In all these cases, we fix a node Y' that is between Y and H on the path p, and prove that if $P_{\mathbf{x}'}(y' \mid \mathbf{w}'')$ is not identifiable, then neither is $P_{\mathbf{x}'}(y \mid \mathbf{w}'')$.

Assume neither Y nor Y' have descendants in $\mathbf{W}''$. If Y' is a parent of Y as in Figure 14 (a), then $P_{\mathbf{x}'}(y \mid \mathbf{w}'') = \sum_{y'} P(y \mid y') P_{\mathbf{x}'}(y' \mid \mathbf{w}'')$. If Y is a parent of Y', as in Figure 14 (b) then the next node in p must be a child of Y'. Therefore, $P_{\mathbf{x}'}(y \mid \mathbf{w}'') = \sum_{y'} P(y \mid y') P_{\mathbf{x}'}(y' \mid \mathbf{w}'')$. In either case, by construction $P(Y \mid Y')$ is a 2 by 2 identity matrix. This implies that the mapping from $P_{\mathbf{x}'}(y' \mid \mathbf{w}'')$ to $P_{\mathbf{x}'}(y \mid \mathbf{w}'')$ is one to one. If Y' and Y share a hidden common parent U as in Figure 15 (b), then our result follows by combining the previous two cases.

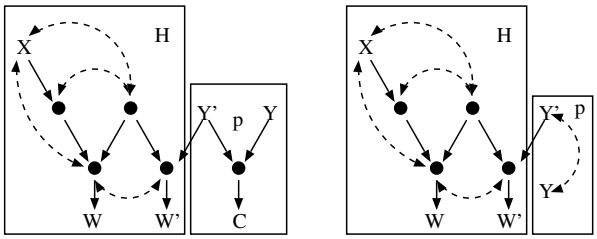


Figure 15: Inductive cases for proving non-identifiability of $P_x(y \mid w, w')$.

The next case is if Y and Y have a common child C which is either in $\mathbf{W}''$ or has a descendant in $\mathbf{W}''$, as in Figure 15 (a). Now $P_{\mathbf{x}'}(y \mid \mathbf{w}'') = \sum_{y'} P(y \mid y', c) P_{\mathbf{x}'}(y' \mid \mathbf{w}'')$. Because all nodes in $\mathbf{W}''$ were observed to be 0, $P(y \mid y', c)$ is again a 2 by 2 identity matrix.

Finally, we handle the base cases of our induction. In all such cases, Y is the first node not in H on the path p. Let Y' be the last node in H on the path p.

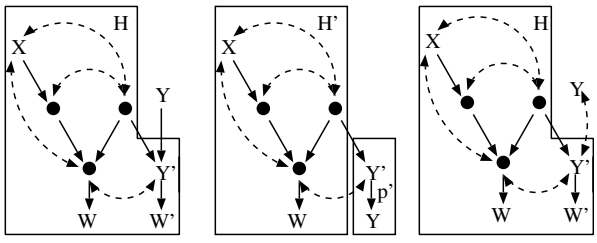


Figure 16: Base cases for proving non-identifiability of $P_x(y \mid w, w')$.

Assume Y is a parent of Y', as shown in Figure 16 (a). By Lemma 41, we can assume $Y \notin An(F \setminus F')_H$. By construction, $(\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U}) \pmod 2$ in M^1_*, and

$(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F'))$ (mod 2) in M^2_*. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U})$ (mod 2) in M^1_*, and $Y = (2 * \sum(\mathbf{U} \cap F'))$ (mod 2) in M^2_*. If an intervention $do(\mathbf{x})$ is performed, $(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F'))$ (mod 2) in $M^2_{*\mathbf{x}}$, by construction. Thus if $\mathbf{W}''$ are all observed to be zero, $Y = 0$ with probability 1. Note that in $M^1_{\mathbf{x}}$ as constructed in the proof of Theorem 17, $(\sum \mathbf{w}'' = \mathbf{x} + \sum \mathbf{U}')$ (mod 2), where $\mathbf{U}' \subseteq \mathbf{U}$ consists of unobservable nodes with one child in $An(\mathbf{X})_F$ and one child in $F \setminus An(\mathbf{X})_F$.

Because $Y \notin An(F \setminus F')_H$, we can conclude that if $\mathbf{W}''$ are observed to be 0, $Y = (\mathbf{x} + \sum \mathbf{U}')$ (mod 2) in $M^1_{*\mathbf{x}'}$. Thus, $Y = 0$ with probability 0.5. Therefore, $P^1_{*\mathbf{x}'}(y \mid \mathbf{w}'') \neq P^2_{*\mathbf{x}'}(y \mid \mathbf{w}'')$ in this case.

Assume Y is a child of Y'. Now consider a graph G' which is obtained from $H \cup p$ by removing the (unique) outgoing arrow from Y' in H. If $P_{\mathbf{x}'}(Y \mid \mathbf{w}'')$ is not identifiable in G', we are done. Assume $P_{\mathbf{x}'}(Y \mid \mathbf{w}'')$ is identifiable in G'. If $Y' \in F$, and $\mathbf{R}$ is the root set of F, then removing the Y'-outgoing directed arrow from F results in a new C-forest, with a root set $\mathbf{R} \cup \{Y'\}$. Because Y is a child of Y', the new C-forests form a hedge for $P_{\mathbf{x}'}(y, \mathbf{w}'')$. If $Y' \in H \setminus F$, then removing the Y'-outgoing directed arrow results in substituting Y for $W \in \mathbf{W}'' \cap De(Y')_H$. Thus in G', F, F' form a hedge for $P_{\mathbf{x}'}(y, \mathbf{w}'' \setminus \{w\})$. In either case, $P_{\mathbf{x}'}(y, \mathbf{w}'')$ is not identifiable in G'.

If $P_{\mathbf{x}'}(\mathbf{w}'')$ is identifiable in G', we are done. If not, consider a smaller hedge $H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. We can repeat the inductive argument for H', p' and Y. See Figure 16 (b).

If $P_{\mathbf{x}'}(\mathbf{w}'')$ is identifiable in G', we are done. If not, consider a smaller hedge $H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. We can repeat the inductive argument for H', p' and Y. See Figure 16 (b). If Y and Y' have a hidden common parent, as is the case in Figure 16 (c), we can combine the first inductive case, and the first base case to prove our result.

We conclude the proof by introducing a slight change to rid us of non-positivity in the distributions P^1, P^2 in our counterexamples. Specifically, for every node I in $p \cup (De(\mathbf{R}) \cap An(\mathbf{Y}))$, add a new binary exogenous parent U_I which is independent of other nodes in $\mathbf{U}$, and has an arbitrarily small probability of assuming the value 1, and causing its child to flip its current value. We let P_{odd} be the probability an odd number of U_I nodes assume the value 1. Because $P(U_I = 1)$ is vanishingly small for every I, P_{odd} is much smaller than 0.5. It's easy to see that P is positive in counterexamples augmented in this way. In the base case when Y is a parent of Y', we modify our equations to account for the addition of U_I. Specifically, $(\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U} + \sum \mathbf{U_I})$ (mod 2) in M^1_*, and $(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F') + \sum \mathbf{U_I})$ (mod 2) in M^2_*, where $U_{\mathbf{U}}$ is the set of nodes added. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U} + \sum \mathbf{U_I})$ (mod 2) in M^1_*, and $Y = (2 * \sum(\mathbf{U} \cap F') + \sum \mathbf{U_I})$ (mod 2) in M^2_*. So prior to the intervention, $P(Y = 1 \mid \mathbf{w}'') = P_{odd}$. But because $P^1_{\mathbf{x}'}(Y = 1 \mid \mathbf{w}'') = 0.5$, adding U_I nodes to the model does not change this probability. Because $P^2(Y = 1 \mid \mathbf{w}'') = P^2_{\mathbf{x}}(Y = 1 \mid \mathbf{w}'')$, our conclusion follows.

In the inductive cases above, we showed that $P_{\mathbf{x}}(Y' = Y \mid \mathbf{W}'') = 1$ in our counterexamples. It's easy to see that with the addition of U_I, $P_{\mathbf{x}}(Y' = Y \mid \mathbf{W}'') = P_{odd}$. This implies that if $P^1_{\mathbf{x}}(Y' \mid \mathbf{W}'') \neq P^2_{\mathbf{x}}(Y' \mid \mathbf{W}'')$, then $P^1_{\mathbf{x}}(Y \mid \mathbf{W}'') \neq P^2_{\mathbf{x}}(Y \mid \mathbf{W}'')$.

This completes the proof. ■

What remains for us to show are the theorems which imply the soundness and completeness results in Section 4. The most important point in these proofs is that

counterfactual graphs are generally no different from causal diagrams discussed in Sections 2 and 3, with their only special feature being that by construction, some nodes in the graph happen to share functions. This means that a lot of results we already proved for Section 3 can be reused without change.

Lemma 42 *If the preconditions of line 7 are met,* $P(S) = P_{\boldsymbol{x}}(\boldsymbol{var}(S))$*, where* $\boldsymbol{x} = \bigcup \boldsymbol{sub}(S)$.

Proof Let $\mathbf{x} = \bigcup \mathbf{sub}(S)$. Since the preconditions are met, $\mathbf{x}$ does not contain conflicting assignments to the same variable, which means $do(\mathbf{x})$ is a sound action in the original causal model. Note that for any variable $Y_{\mathbf{w}}$ in S, any variable in $(Pa(S) \setminus S) \cap An(Y_{\mathbf{w}})_S$ is already in $\mathbf{w}$, while any variable in $(Pa(S) \setminus S) \setminus An(Y_{\mathbf{w}})_S$ can be added to the subscript of $Y_{\mathbf{w}}$ without changing the variable. Since $Y \cap \mathbf{X} = \emptyset$ by assumption, $Y_{\mathbf{w}} = Y_{\mathbf{x}}$. Since $Y_{\mathbf{w}}$ was arbitrary, our result follows. ■

For convenience, we show the soundness of **ID*** and **IDC*** asserted in Theorem 26 separately.

Theorem 26 (a) *If* ***ID**** *succeeds, the expression it returns is equal to* $P(\gamma)$ *in a given causal graph.*

Proof The proof outline in Section 3 is sufficient for everything except the base cases. In particular, line 6 follows by Lemma 33. For soundness, we only need to handle the positive base case, which follows from Lemma 42. ■

The soundness of **IDC*** is also fairly straightforward.

Theorem 26 (b) *If* ***IDC**** *does not output* ***FAIL****, the expression it returns is equal to* $P(\gamma \mid \delta)$ *in a given causal graph, if that expression is defined, and* ***UNDEFINED*** *otherwise.*

Proof Theorem 20 shows how an operation similar to line 4 is sound by rule 2 of do-calculus (Pearl, 1995) when applied in a causal diagram. But we know that the counterfactual graph is just a causal diagram for a model where some nodes share functions, so the same reasoning applies. The rest is straightforward. ■

To show completeness of **ID*** and **IDC***, we first prove a utility lemma which will make it easier to construct counterexamples which agree on P_* but disagree on a given counterfactual query.

Lemma 43 *Let* G *be a causal graph partitioned into a set* $\{S_1, \ldots, S_k\}$ *of C-components. Then two models* M_1, M_2 *which induce* G *agree on* P_* *if and only if their submodels* $M^1_{\boldsymbol{v}\setminus s_i}, M^2_{\boldsymbol{v}\setminus s_i}$ *agree on* P_* *for every C-component* S_i*, and value assignment* $\boldsymbol{v} \setminus s_i$.

Proof This follows from C-component factorization: $P(\mathbf{v}) = \prod_i P_{\mathbf{v}\setminus s_i}(s_i)$. This implies that for every $do(\mathbf{x})$, $P_{\mathbf{x}}(\mathbf{v})$ can be expressed as a product of terms $P_{\mathbf{v}\setminus(s_i\setminus\mathbf{x})}(s_i \setminus \mathbf{x})$, which implies the result. ■

The next result generalizes Lemma 27 to a wider set of counterfactual graphs which result from non-identifiable queries.

Lemma 28 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1, \ldots, W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1, \ldots, w^k, z_{x'}), P(y_x, w^1, \ldots, w^k, z)$ for any value assignments $y, w^1, \ldots, w^k, z$.*

Proof We construct two models with graph G as follows. In both models, all variables are binary, and $P(\mathbf{u})$ is uniform. In M^1, each variable is set to the bit parity of its parents. In M^2, the same is true except Y and Z ignore the values of X. To prove that the two models agree on P_*, we use Lemma 43. Clearly the two models agree on $P(X)$. To show that the models also agree on $P_x(\mathbf{V} \setminus X)$ for all values of x, note that in M_2 each value assignment over $\mathbf{V} \setminus X$ with even bit parity is equally likely, while no assignment with odd bit parity is possible. But the same is true in M^1 because any value of x contributes to the bit parity of $\mathbf{V} \setminus X$ exactly twice. The agreement of M^1_x, M^2_x on P_* follows by the graph structure of G.

To see that the result is true, we note firstly that $P(\Sigma_i W^i + Y_x + Z_{x'} \pmod 2 = 1) = P(\Sigma_i W^i + Y_x + Z \pmod 2 = 1) = 0$ in M^2, while the same probabilities are positive in M^1, and secondly that in both models distributions $P(y_x, w^1, \ldots, w^k, z_{x'})$ and $P(y_x, w^1, .., w^k, z)$ assign equal probabilities to outcomes with positive probabilities, while we just established that the set of these possible outcomes differs in M_1 and M_2. Note that the proof is easy to generalize for positive P_* by adding a small probability for Y to flip its normal value. ■

To obtain a full characterization of non-identifiable counterfactual graphs, we augment the difficult graphs we obtained from the previous two results using certain graph transformation rules which preserve non-identifiability. These rules are given in the following two lemmas.

Lemma 29 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let $\{y^1_{\mathbf{x}^1}, \ldots, y^n_{\mathbf{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, \ldots, Y^n$, and let P'_* be the set of all interventional distributions in models inducing G'. Let $\gamma' = (\gamma \setminus \{y^1_{\mathbf{x}^1}, \ldots, y^n_{\mathbf{x}^m}\}) \cup \{w_{\mathbf{x}^1}, \ldots, w_{\mathbf{x}^m}\}$, where w is an arbitrary value of W. Then $P'_*, G' \not\vdash_{id} P(\gamma')$.*

Proof Let M^1, M^2 witness $P_*, G \not\vdash_{id} P(\gamma)$. We will extend these models to witness $P'_*, G' \not\vdash_{id} P(\gamma')$. Since the function of a newly added W will be shared, and M^1, M^2 agree on P_* in G, the extensions will agree on P'_* by Lemma 43. We have two cases.

Assume there is a variable Y^i such that $y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k}$ are in γ. By Lemma 27, $P_*, G \not\vdash_{id} P(y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k})$. Then let W be a child of just Y^i, and assume $|W| = |Y^i| = c$. Let W be set to the value of Y^i with probability $1 - \epsilon$, and otherwise it is set to a uniformly chosen random value of Y^i among the other $c - 1$ values. Since ϵ is arbitrarily small, and since $W_{\mathbf{x}^j}$ and $W_{\mathbf{x}^k}$ pay attention to the same U variable, it is possible to set ϵ in such a way that if $P^1(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k}) \neq P^2(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k})$, however minutely, then $P^1(W_{\mathbf{x}^j}, W_{\mathbf{x}^k}) \neq P^2(W_{\mathbf{x}^j}, W_{\mathbf{x}^k})$.

Otherwise, let $|W| = \prod_i |Y^i|$, and let $P(W \mid Y^1, \ldots, Y^n)$ be an invertible stochastic matrix. Our result follows. ■

Lemma 30 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If* $(\exists w_{\boldsymbol{s}} \in \gamma)$ *where* $x \in \boldsymbol{s}$, $y \notin \boldsymbol{s}$, *and* $X \in An(W)_G$, *then* $Y \notin An(W)_G$.
- *If* $(\exists y_{\boldsymbol{s}} \in \gamma)$ *where* $x \in \boldsymbol{s}$, *then* $An(X)_G = \emptyset$.
- *If* $(Y_{\boldsymbol{w}}, X_{\boldsymbol{s}} \in \gamma)$, *then* $\boldsymbol{w}$ *and* $\boldsymbol{s}$ *agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ *and there's some isomorphism* f *assigning value pairs* x, y *to a value* $f(x, y) = z$. *Let* γ' *be obtained from* γ *as follows. For any* $w_{\boldsymbol{s}} \in \gamma$:

- *If* $W \notin \{X, Y\}$, *and values* x, y *occur in* $\boldsymbol{s}$, *replace them by* $f(x, y)$.
- *If* $W \notin \{X, Y\}$, *and the value of one of* X, Y *occur in* $\boldsymbol{s}$, *replace it by some* z *consistent with the value of* X *or* Y.
- *If* X, Y *do not occur in* γ, *leave* γ *as is.*
- *If* $W = Y$ *and* $x \in \boldsymbol{s}$, *replace* $w_{\boldsymbol{s}}$ *by* $f(x, y)_{\boldsymbol{s} \setminus \{x\}}$.
- *otherwise, replace every variable pair of the form* $Y_{\boldsymbol{r}} = y, X_{\boldsymbol{s}} = x$ *by* $Z_{\boldsymbol{r},\boldsymbol{s}} = f(x, y)$.

Then $P_*, G' \not\vdash_{id} P(\gamma')$.

Proof Let Z be the Cartesian product of X, Y, and fix f. We want to show that the proof of non-identification of $P(\gamma)$ in G carries over to $P(\gamma')$ in G'.

We have five modification conditions which can apply to a variable $w_{\mathbf{s}} \in \gamma$. However, since γ is left alone if X, Y do not occur in γ (the third condition), only the remaining four of these conditions result in an actual modification of a counterfactual variable in γ.

We go through these remaining conditions one by one. The first clearly results in the same counterfactual variable. For the second, due to the restrictions we imposed, $w_{\mathbf{z}} = w_{\mathbf{z},y,x}$, which means we can apply the first modification.

For the fourth, we have $P(\gamma) = P(\delta, y_{x,\mathbf{z}})$. By our restrictions, and rule 2 of do-calculus (Pearl, 1995), this is equal to $P(\delta, y_{\mathbf{z}} \mid x_{\mathbf{z}})$. Since this is not identifiable, then neither is $P(\delta, y_{\mathbf{z}}, x_{\mathbf{z}})$. Now it's clear that our modification is equivalent to one applied after the fifth condition.

The fifth modification is simply a merge of events consistent with a single causal world into a conjunctive event, which does not change the overall expression. ■

We are now ready to show the main completeness results for counterfactual identification algorithms. Again, we prove this results separately for **ID*** and **IDC*** for convenience.

Theorem 31 (a) ***ID**** *is complete.*

Proof We want to show that if line 8 fails, the original $P(\gamma)$ cannot be identified. There are two broad cases to consider. If G_γ contains the w-graph, the result follows by Lemmas 27 and 29. If not, we argue as follows.

Fix some X which witnesses the precondition on line 8. We can assume X is a parent of some nodes in S. Assume no other node in $\mathbf{sub}(S)$ affects S (effectively we delete all

edges from parents of S to S except from X). Because the w-graph is not a part of G_γ, this has no ramifications on edges in S. Further, we assume X has two values in S.

If $X \notin S$, fix $Y, W \in S \cap Ch(X)$. Assume S has no directed edges at all. Then $P_*, G \not\vdash_{id} P(S)$ by Lemma 28. The result now follows by Lemma 29, and by construction of G_γ, which implies all nodes in S have some descendant in γ.

If S has directed edges, we want to show $P_*, G \not\vdash_{id} P(R(S))$, where $R(S)$ is the subset of S with no children in S. We can recover this from the previous case as follows. Assume S has no edges as before. For a node $Y \in S$, fix a set of childless nodes $\mathbf{X} \in S$ which are to be their parents. Add a virtual node Y' which is a child of all nodes in $\mathbf{X}$. Then $P_*, G \not\vdash_{id} P((S \setminus \mathbf{X}) \cup Y')$ by Lemma 29. Then $P_*, G \not\vdash_{id} P(R(S'))$, where S' is obtained from S by adding edges from $\mathbf{X}$ to Y by Lemma 30, which applies because no w-graph exists in G_γ. We can apply this step inductively to obtain the desired forest (all nodes have at most one child) S while making sure $P_*, G \not\vdash_{id} P(R(S))$.

If S is not a forest, we can simply disregard extra edges so effectively it is a forest. Since the w-graph is not in G_γ this does not affect edges from X to S.

If $X \in S$, fix $Y \in S \cap Ch(X)$. If S has no directed edges at all, replace X by a new virtual node Y, and make X be the parent of Y. By Lemma 28, $P_*, G \not\vdash_{id} P((S \setminus x) \cup y_x)$. We now repeat the same steps as before, to obtain that $P_*, G \not\vdash_{id} P((R(S) \setminus x) \cup y_x)$ for general S. Now we use Lemma 30 to obtain $P_*, G \not\vdash_{id} P(R(S))$. Having shown $P_*, G \not\vdash_{id} P(R(S))$, we conclude our result by inductively applying Lemma 29. ■

Theorem 31 (b) ***IDC**** *is complete.*

Proof The difficult step is to show that after line 5 is reached, if $P_*, G \not\vdash_{id} P(\gamma, \delta)$ then $P_*, G \not\vdash_{id} P(\gamma \mid \delta)$. If $P_*, G \vdash_{id} P(\delta)$, this is obvious. Assume $P_*, G \not\vdash_{id} P(\delta)$. Fix the S which witnesses that for $\delta' \subseteq \delta$, $P_*, G \not\vdash_{id} P(\delta')$. Fix some Y such that a back-door, that is, starting with an incoming arrow, path exists from δ' to Y in $G_{\gamma,\delta}$. We want to show that $P_*, G \not\vdash_{id} P(Y \mid \delta')$. Let $G' = G_{An(\delta') \cap De(S)}$.

Assume Y is a parent of a node $D \in \delta'$, and $D \in G'$. Augment the counterexample models which induce counterfactual graph G' with an additional binary node for Y, and let the value of D be set as the old value plus Y modulo $|D|$. Let Y attain value 1 with vanishing probability ϵ. That the new models agree on P_* is easy to establish. To see that $P_*, G \not\vdash_{id} P(\delta')$ in the new model, note that $P(\delta')$ in the new model is equal to $P(\delta' \setminus D, D = d) * (1 - \epsilon) + P(\delta' \setminus D, D = (d-1) \pmod{|D|}) * \epsilon$. Because ϵ is arbitrarily small, this implies our result. To show that $P_*, G \not\vdash_{id} P(Y = 1 \mid \delta')$, we must show that the models disagree on $P(\delta' \mid Y = 1)/P(\delta')$. But to do this, we must simply find two consecutive values of D, $d, d+1 \pmod{|D|}$ such that $P(\delta' \setminus D, d+1 \pmod{|D|})/P(\delta' \setminus D, d)$ is different in the two models. But this follows from non-identification of $P(\delta')$.

If Y is not a parent of $D \in G'$, then either it is further along on the back-door path or it's a child of some node in G'. In case 1, we must construct the distributions along the back-door path in such a way that if $P_*, G \not\vdash_{id} P(Y' \mid \delta')$ then $P_*, G \not\vdash_{id} P(Y \mid \delta')$, where Y' is a node preceding Y on the path. The proof follows closely the one in Theorem 21. In case 2, we duplicate the nodes in G' which lead from Y to δ', and note that we can show non-identification in the resulting graph using reasoning in case 1. We obtain our result by applying Lemma 30. ■

References

Chen Avin, Ilya Shpitser, and Judea Pearl. Identifiability of path-specific effects. In *International Joint Conference on Artificial Intelligence*, volume 19, pages 357–363, 2005.

Alexander Balke and Judea Pearl. Counterfactual probabilities: Computational methods, bounds and applications. In *Proceedings of UAI-94*, pages 46–54, 1994a.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *Proceedings of AAAI-94*, pages 230–237, 1994b.

Alexander Philip Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society*, 41:1–31, 1979.

David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundation of Science*, 3:151–182, 1998.

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

Joseph Halpern. Axiomatizing causal reasoning. *Journal of A.I. Research*, pages 317–337, 2000.

Yimin Huang and Marco Valtorta. Pearl's calculus of interventions is complete. In *Twenty Second Conference On Uncertainty in Artificial Intelligence*, 2006a.

Yimin Huang and Marco Valtorta. Identifiability in causal bayesian networks: A sound and complete algorithm. In *Twenty-First National Conference on Artificial Intelligence*, 2006b.

Manabu Kuroki and Masami Miyakawa. Identifiability criteria for causal effects of joint interventions. *Journal of Japan Statistical Society*, 29:105–117, 1999.

Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan and Kaufmann, San Mateo, 1988.

Judea Pearl. Graphical models, causality, and intervention. *Statistical Science*, 8:266–9, 1993a.

Judea Pearl. A probabilistic calculus of actions. In *Uncertainty in Artificial Intelligence (UAI)*, volume 10, pages 454–462, 1993b.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995. URL http://citeseer.ist.psu.edu/55450.html.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000. ISBN 0-521-77362-8.

Judea Pearl. Direct and indirect effects. In *Proceedings of UAI-01*, pages 411–420, 2001.

Judea Pearl and James M. Robins. Probabilistic evaluation of sequential plans from causal models with hidden variables. In *Uncertainty in Artificial Intelligence*, volume 11, pages 444–453, 1995.

Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Twenty-First National Conference on Artificial Intelligence*, 2006a.

Ilya Shpitser and Judea Pearl. Identification of conditional interventional distributions. In *Uncertainty in Artificial Intelligence*, volume 22, 2006b.

Ilya Shpitser and Judea Pearl. What counterfactuals can be tested. In *Twenty Third Conference on Uncertainty in Artificial Intelligence, forthcoming*. Morgan Kaufmann, 2007.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Springer Verlag, New York, 1993.

Jin Tian. *Studies in Causal Reasoning and Learning*. PhD thesis, Department of Computer Science, University of California, Los Angeles, 2002.

Thomas S. Verma. Causal networks: semantics and expressiveness. Technical Report R-65, Cognitive Systems Laborator, University of California, Los Angeles, 1986.

Sewall Wright. Correlation and causation. *Journal of Agricultural Research*, 20:557–585, 1921.

JMLR: Workshop and Conference Proceedings 9:2523–2547, 2008 Submitted 6/07; Revised 2/08; Published 11/08

Active Learning of Causal Networks with Intervention Experiments and Optimal Designs

Yang-Bo He HEYB@MATH.PKU.EDU.CN
Zhi Geng ZGENG@MATH.PKU.EDU.CN
School of Mathematical Sciences, LMAM
Peking University
Beijing 100871, China

Editor: André Elisseeff

Abstract

The causal discovery from data is important for various scientific investigations. Because we cannot distinguish the different directed acyclic graphs (DAGs) in a Markov equivalence class learned from observational data, we have to collect further information on causal structures from experiments with external interventions. In this paper, we propose an active learning approach for discovering causal structures in which we first find a Markov equivalence class from observational data, and then we orient undirected edges in every chain component via intervention experiments separately. In the experiments, some variables are manipulated through external interventions. We discuss two kinds of intervention experiments, randomized experiment and quasi-experiment. Furthermore, we give two optimal designs of experiments, a batch-intervention design and a sequential-intervention design, to minimize the number of manipulated variables and the set of candidate structures based on the minimax and the maximum entropy criteria. We show theoretically that structural learning can be done locally in subgraphs of chain components without need of checking illegal v-structures and cycles in the whole network and that a Markov equivalence subclass obtained after each intervention can still be depicted as a chain graph.

Keywords: active learning, causal networks, directed acyclic graphs, intervention, Markov equivalence class, optimal design, structural learning

1. Introduction

A directed acyclic graph (DAG) (also called a Bayesian network) is a powerful tool to describe a large complex system in various scientific investigations, such as bioinformatics, epidemiology, sociology and business (Pearl, 1988; Lauritzen, 1996; Whittaker, 1990; Aliferis et al., 2003; Jansen et al., 2003; Friedman, 2004). A DAG is also used to describe causal relationships among variables. It is crucial to discover the structure of a DAG for understanding a large complex system or for doing uncertainty inference on it (Cooper and Yoo, 1999; Pearl, 2000). There are many methods of structural learning, and the main methods are Bayesian methods (Cooper and Yoo, 1999; Heckerman, 1997) and constraint-based methods (Spirtes et al., 2000). From data obtained in observational studies, we may not have enough information to discover causal structures completely, but we can obtain only a Markov equivalence class. Thus we have to collect further information of causal structures via experiments with external interventions. Heckerman et al. (1995) discussed structural learning of Bayesian networks from a combination

of prior knowledge and statistical data. Cooper and Yoo (1999) presented a method of causal discovery from a mixture of experimental and observational data. Tian and Pearl (2001a,b) proposed a method of discovering causal structures based on dynamic environment. Tong and Koller (2001) and Murphy (2001) discussed active learning of Bayesian network structures with posterior distributions of structures based on decision theory. In these methods, causal structures are discovered by using additional information from domain experts or experimental data.

Chain graphs were introduced as a natural generalization of DAGs to admit more flexible causal interpretation (Lauritzen and Richardson, 2002). A chain graph contains both directed and undirected edges. A chain component of a chain graph is a connected undirected graph obtained by removing all directed edges from the chain graph. Andersson et al. (1997) showed that DAGs in a Markov equivalence class can be represented by a chain graph. He et al. (2005) presented an approach of structural learning in which a Markov equivalence class of DAGs is sequentially refined into some smaller subclasses via domain knowledge and randomized experiments.

In this paper, we discuss randomized experiments and quasi-experiments of external interventions. We propose a method of local orientations in every chain component, and we show theoretically that the method of local orientations does not create any new v-structure or cycle in the whole DAG provided that neither v-structure nor cycle is created in any chain component. Thus structural learning can be done locally in every chain component without need of checking illegal v-structures and cycles in the whole network. Then we propose the optimal designs of interventional experiments based on the minimax and maximum entropy criteria. These results greatly extend the approach proposed by He et al. (2005). In active learning, we first find a Markov equivalence class from observational data, which can be represented by a chain graph, and then we orient undirected edges via intervention experiments. Two kinds of intervention experiments can be used for orientations. One is randomized experiment, in which an individual is randomly assigned to some level combination of the manipulated variables at a given probability. Randomization can disconnect the manipulated variables from their parent variables in the DAG. Although randomized experiments are most powerful for learning causality, they may be inhibitive in practice. The other is quasi-experiment, in which the pre-intervention distributions of some variables are changed via external interventions, but we cannot ensure that the manipulated variables can be disconnected from their parent variables in the DAG, and thus the post-intervention distributions of manipulated variables may still depend on their parent variables. For example, the pre-intervention distribution of whether patients take a vaccine or not may depend on some variables, and the distribution may be changed by encouraging patients with some benefit in the quasi-experiment, but it may still depend on these variables. Furthermore, we discuss the optimal designs by which the number of manipulated variables is minimized or the uncertainty of candidate structures is minimized at each experiment step based on the minimax and the maximum entropy criteria. We propose two kinds of optimal designs: a batch-intervention experiment and a sequential intervention experiment. For the former, we try to find the minimum set of variables to be manipulated in a batch such that undirected edges are all oriented after the interventions. For the latter, we first choose a variable to be manipulated such that the Markov equivalence class can be reduced by manipulating the variable into a subclass as small as possible, and then according to the current subclass, we repeatedly choose a next variable to be manipulated until all undirected edges are oriented.

In Section 2, we introduce notation and definitions and then show some theoretical results on Markov equivalence classes. In Section 3, we present active learning of causal structures via external interventions and discuss randomized experiments and quasi-experiments. In Section 4, we propose two optimal designs of intervention experiments, a batch-intervention design and a sequential intervention design. In Section 5, we show simulation results to evaluate the performances of intervention designs proposed in this paper. Conclusions are given in Section 6. Proofs of theorems are given in Appendix A.

2. Causal DAGs and Markov Equivalence Class

A graph G can be defined to be a pair $G = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{V_1, \cdots, V_n\}$ denotes the node set and $\mathbb{E}$ denotes the edge set which is a subset of the set $\mathbb{V} \times \mathbb{V}$ of ordered pairs of nodes. If both ordered pairs (V_i, V_j) and (V_j, V_i) are in $\mathbb{E}$, we say that there is an undirected edge between V_i and V_j, denoted as $V_i - V_j$. If $(V_i, V_j) \in \mathbb{E}$ and $(V_j, V_i) \notin \mathbb{E}$, we call it a directed edge, denoted as $V_i \to V_j$. We say that V_i is a neighbor of V_j if there is an undirected or directed edge between V_i and V_j. A graph is directed if all edges of the graph are directed. A graph is undirected if all edges of the graph are undirected.

A sequence $(V_1, V_2, \cdots, V_k)$ is called a *partially directed path* from V_1 to V_k if either $V_i \to V_{i+1}$ or $V_i - V_{i+1}$ is in G for all $i = 1, \ldots, k-1$. A partially directed path is a directed path if there is not any undirected edge in the path. A node V_i is an *ancestor* of V_j and V_j is a *descendant* of V_i if there is a directed path from V_i to V_j. A *directed cycle* is a directed path from a node to itself, and a *partially directed cycle* is a partially directed path from a node to itself.

A graph with both directed and undirected edges is a chain graph if there is not any partially directed cycle. Figure 1 shows a chain graph with five nodes. A chain component is a node set whose nodes are connected in an undirected graph obtained by removing all directed edges from the chain graph. An undirected graph is chordal if every cycle of length larger than or equal to 4 possesses a chord.

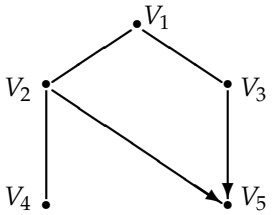


Figure 1: A chain graph G^* depicts the essential graph of G, G_1, G_2 and G_3.

A directed acyclic graph (DAG) is a directed graph which does not contain any directed cycle. A causal DAG is a DAG which is used to describe the causal relationships among variables $V_1, \cdots, V_n$. In the causal DAG, a directed edge $V_i \to V_j$ is interpreted as that the *parent* node V_i is a cause of the *child* node V_j, and that V_j is an effect of V_i. Let $pa(V_i)$ denote the set of all parents of V_i and $ch(V_i)$ denote the set of all *children* of V_i. Let τ be a node subset of $\mathbb{V}$. The *subgraph* $G_\tau = (\tau, \mathbb{E}_\tau)$ induced by the subset τ has the node set τ and the edge set $\mathbb{E}_\tau = \mathbb{E} \cap (\tau \times \tau)$ which contains all edges falling into τ.

Two graphs have *the same skeleton* if they have the same set of nodes and the same set of edges regardless of their directions. A head-to-head structure is called a *v-structure* if the parents are not adjacent, such as $V_1 \rightarrow V_2 \leftarrow V_3$.

Figure 2 shows four different causal structures of five nodes. The causal graph G in Figure 2 depicts that V_1 is a cause of V_3, which in turn is a cause of V_5.

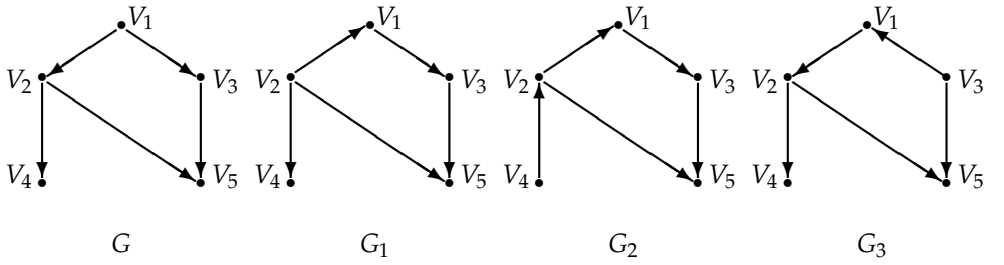


Figure 2: The equivalence class $[G]$.

A joint distribution P satisfies Markov property with respect to a graph G if any variable of G is independent of all its non-descendants in G given its parents with respect to the joint distribution P. Furthermore, the distribution P can be factored as follows

$$P(v_1, v_2, \cdots, v_n) = \prod_{i=1}^{n} P(v_i \mid pa(v_i)),$$

where v_i denotes a value of variable V_i, and $pa(v_i)$ denotes a value of the parent set $pa(V_i)$ (Pearl, 1988; Lauritzen, 1996; Spirtes et al., 2000). In this paper, we assume that any conditional independence relations in P are entailed by the Markov property, which is called the faithfulness assumption (Spirtes et al., 2000). We also assume that there are no latent variables (that is, no unmeasured variables) in causal DAGs. Different DAGs may encode the same Markov properties. A Markov equivalence class is a set of DAGs that have the same Markov properties. Let $G_1 \sim G_2$ denote that two DAGs G_1 and G_2 are Markov equivalent, and let $[G]$ denote the equivalence class of a DAG G, that is, $[G] = \{G' : G' \sim G\}$. The four DAGs G, G_1, G_2 and G_3 in Figure 2 form a Markov equivalence class $[G]$. Below we review two results about Markov equivalence of DAGs given by Verma and Pearl (1990) and Andersson et al. (1997).

Lemma 1 *(Verma and Pearl, 1990) Two DAGs are Markov equivalent if and only if they have the same skeleton and the same v-structures.*

Andersson et al. (1997) used an essential graph G^* to represent the equivalence class $[G]$.

Definition 2 *The essential graph $G^* = (\mathbb{V}, \mathbb{E}^*)$ of G has the same node set and the same skeleton as G, whose one edge is directed if and only if it has the same orientation in every DAG in $[G]$ and whose other edges are undirected.*

For example, G^* in Figure 1 is the essential graph of G in Figure 2. The edges $V_2 \rightarrow V_5$ and $V_3 \rightarrow V_5$ in G^* are directed since they have the same orientation for all DAGs of $[G]$ in Figure 2, and other edges are undirected.

Lemma 3 *(Andersson et al., 1997) Let G^* be the essential graph of $G = (\mathbb{V}, \mathbb{E})$. Then G^* has the following properties:*

(i) *G^* is a chain graph,*

(ii) *G^*_τ is chordal for every chain component τ, and*

(iii) *$V_i \to V_j - V_k$ does not occur as an induced subgraph of G^*.*

Suppose that G is an unknown underlying causal graph and that its essential graph $G^* = (\mathbb{V}, \mathbb{E})$ has been obtained from observational data, and has k chain components $\{\tau_1, \cdots, \tau_k\}$. Its edge set $\mathbb{E}$ can be partitioned into the set $\mathbb{E}_1$ of directed edges and the set $\mathbb{E}_2$ of undirected edges. Let G^*_τ denote a subgraph of the essential G^* induced by a chain component τ of G^*. Any subgraph of the essential graph induced by a chain component is undirected. Since all v-structures can be discovered from observational data, any subgraph G'_τ of G' should not have any v-structure for $G' \in [G]$. For example, the essential graph G^* in Figure 1 has one chain component $\tau = \{V_1, V_2, V_3, V_4\}$. It can been seen that G'_τ has no v-structure for $G' \in \{G, G_1, G_2, G_3\}$.

Given an essential graph G^*, we need to orient all undirected edges in each chain component to discover the whole causal graph G. Below we show that the orientation can be done separately in every chain component. We also show that there are neither new v-structures nor cycles in the whole graph as long as there are neither v-structures nor cycles in any chain component. Thus in the orientation process, we only need to ensure neither v-structures nor cycles in any component, and we need not check new v-structures and cycles for the whole graph.

Theorem 4 *Let τ be a chain component of an essential graph G^*. For each undirected edge $V - U$ in G^*_τ, neither orientation $V \to U$ nor $V \leftarrow U$ can create a v-structure with any node W outside τ, that is, neither $V \to U \leftarrow W$ nor $W \to V \leftarrow U$ can occur for any $W \notin \tau$.*

Theorem 4 means that there is not any node W outside the component τ which can build a v-structure with two nodes in τ.

Theorem 5 *Let τ be a chain component of G^*. If orientation of undirected edges in the subgraph G^*_τ does not create any directed cycle in the subgraph, then the orientation does not create any directed cycle in the whole DAG.*

According to Theorems 4 and 5, we find that the undirected edges can be oriented separately in each chain component regardless of directed and undirected edges in other part of the essential graph as long as neither cycles nor v-structures are constructed in any chain component. Thus the orientation for one chain component does not affect the orientations for other components. The orientation approach and its correctness will be discussed in Section 3.

3. Active Learning of Causal Structures via External Interventions

To discover causal structures further from a Markov equivalence class obtained from observational data, we have to perform external interventions on some variables. In this section, we consider two kinds of external interventions. One is the randomized experiment, in which the post-intervention distribution of the manipulated variable V_i is independent of its parent variables. The other is the quasi-experiment, in which the

distribution of the manipulated variable V_i conditional on its parents $pa(V_i)$ is changed by manipulating V_i. For example, the distribution of whether patients take a vaccine or not is changed by randomly encouraging patients at a discount.

3.1. Interventions by Randomized Experiments

In this subsection, we conduct interventions as randomized experiments, in which some variables are manipulated from external interventions by assigning individuals to some levels of these variables in a probabilistic way. For example, in a clinical trial, every patient is randomly assigned to a treatment group of $V_i = v_i$ at a probability $P'(v_i)$. The randomized manipulation disconnects the node V_i from its parents $pa(V_i)$ in the DAG. Thus the pre-intervention conditional probability $P(v_i \mid pa(v_i))$ of $V_i = v_i$ given $pa(V_i) = pa(v_i)$ is replaced by the post-intervention probability $P'(v_i)$ while all other conditional probabilities $P(v_j \mid pa(v_j))$ for $j \neq i$ are kept unchanged in the randomized experiment. Then the post-intervention joint distribution is

$$P_{V_i}(v_1, v_2, \cdots, v_n) = P'(v_i) \prod_{j \neq i} P(v_j \mid pa(v_j)),$$

(Pearl, 1993). From this post-intervention distribution, we have $P_{V_i}(v_i \mid pa(v_i)) = P_{V_i}(v_i)$, that is, the manipulated variable V_i is independent of its parents $pa(V_i)$ in the post-intervention distribution. Under the faithfulness assumption, it is obvious that an undirected edge between V_i and its neighbor V_j can be oriented as $V_i \leftarrow V_j$ if the post-intervention distribution has $V_i \perp\!\!\!\perp V_j$, otherwise it is oriented as $V_i \rightarrow V_j$, where $V_i \perp\!\!\!\perp V_j$ denotes that V_i is independent of V_j. The orientation only needs an independence test for the marginal distribution of variables V_i and V_j. Notice that the independence is tested by using only the experimental data without use of the previous observational data.

Let $e(V_i)$ denote the orientation of edges which is determined by manipulating node V_i. If V_i belongs to a chain component τ (that is, it connects at least one undirected edge), then the Markov equivalence class $[G]$ can be reduced by manipulating V_i to the post-intervention Markov equivalence class $[G]_{e(V_i)}$

$$[G]_{e(V_i)} = \{G' \in [G] \mid G' \text{ has the same orientation as } e(V_i)\}.$$

A Markov equivalence class is split into several subclasses by manipulating V_i, each of which has different orientations $e(V_i)$. Let $G^*_{e(V_i)}$ denote the post-intervention essential graph which depicts the post-intervention Markov equivalence class $[G]_{e(V_i)}$. We show below that $G^*_{e(V_i)}$ also has the properties of essential graphs.

Theorem 6 *Let τ be a chain component of the pre-intervention essential graph G^* and V_i be a node in the component τ. The post-intervention graph $G^*_{e(V_i)}$ is also a chain graph, that is, $G^*_{e(V_i)}$ has the following properties:*

(i) *$G^*_{e(V_i)}$ is a chain graph,*

(ii) *$G^*_{e(V_i)}$ is chordal, and*

(iii) *$V_j \rightarrow V_k - V_l$ does not occur as an induced subgraph of $G^*_{e(V_i)}$.*

By Theorem 6, the pre-intervention chain graph is changed by manipulating a variable to another chain graph which has less undirected edges. Thus variables in chain components can be manipulated repeatedly until the Markov equivalence subclass is reduced to a subclass with a single DAG, and properties of chain graphs are not lost in this intervention process.

According to the above results, we first learn an essential graph from observational data, which is a chain graph (Andersson et al., 1997) and depicts a Markov equivalence class (Heckerman et al., 1995; Verma and Pearl, 1990; Castelo and Perlman, 2002). Next we choose a variable V_i to be manipulated from a chain component, and we can orient the undirected edges connecting V_i and some other undirected edges whose reverse orientations create v-structures or cycles. Repeating this process, we choose a next variable to be manipulated until all undirected edges are oriented. Below we give an example to illustrate the intervention process.

Example 1 Consider an essential graph in Figure 3, which depicts a Markov equivalence class with 12 DAGs in Figure 4. After obtaining the essential graph from observational data, we manipulate some variables in randomized experiments to identify a causal structure in the 12 DAGs. For example, Table 1 gives four possible orientations and Markov equivalence subclasses obtained by manipulating V_1. A class with 12 DAGs is split into four subclasses by manipulating V_1. The post-intervention subclasses (ii) and (iv) have only a single DAG separately. Notice that undirected edges not connecting V_1 can also be oriented by manipulating V_1. The subclasses (i) and (iii) are depicted by post-intervention essential graphs (a) and (b) in Table 1 respectively, both of which are chain graphs. In Table 2, the first column gives four possible independence sets obtained by manipulating V_1. For the set with $V_1 \perp\!\!\!\perp V_2$ and $V_1 \not\perp\!\!\!\perp V_3$, the causal structure is the DAG (3) in Figure 4, and thus we need not further manipulate other variables. For the third set with $V_1 \not\perp\!\!\!\perp V_2$ and $V_1 \not\perp\!\!\!\perp V_3$, we manipulate the next variable V_2. If $V_2 \perp\!\!\!\perp V_3$, then the causal structure is the DAG (1), otherwise it is the DAG (2). For the fourth set with $V_1 \perp\!\!\!\perp V_2$ and $V_1 \perp\!\!\!\perp V_3$, we may need further to manipulate variables V_2, V_3 and V_4 to identify a causal DAG.

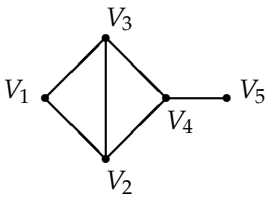


Figure 3: An essential graph of DAGs

3.2. Interventions by Quasi-experiments

In the previous subsection we discussed interventions by randomized experiments. Although randomized experiments are powerful tools to discover causal structures, it may be inhibitive or impractical. In this subsection we consider quasi-experiments. In a quasi-experiment, individuals may choose treatments non-randomly, but their behaviors of treatment choices are influenced by experimenters. For example, some

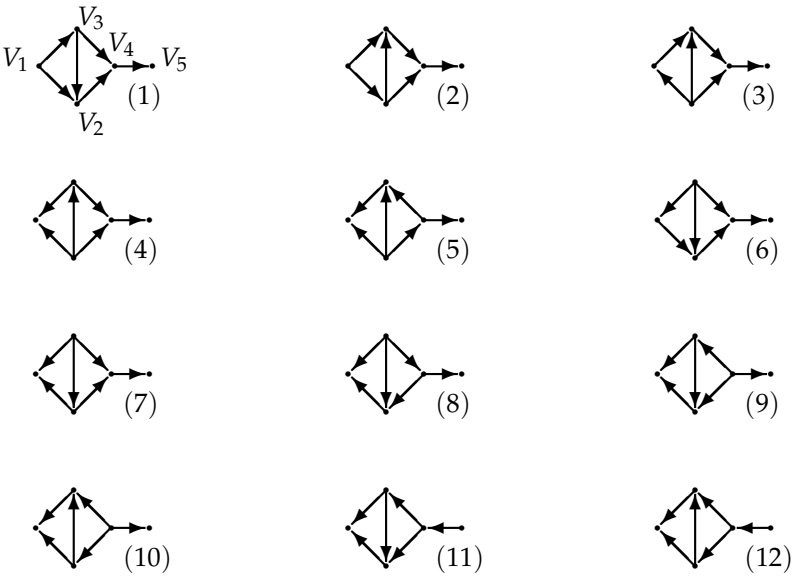


Figure 4: All DAGs in the equivalence class given in Figure 3.

Table 1: The post-intervention subclasses and essential graphs obtained by manipulating V_1.

No of subclass	$e(V_1)$	DAGs in a subclass	post-intervention essential graphs
(i)	$V_2 \leftarrow V_1 \rightarrow V_3$	$(1,2)$	V_1 (a)
(ii)	$V_2 \rightarrow V_1 \rightarrow V_3$	(3)	
(iii)	$V_2 \rightarrow V_1 \leftarrow V_3$	$(4,5,$ $7-12)$	V_1 (b)
(iv)	$V_2 \leftarrow V_1 \leftarrow V_3$	(6)	

patients may not comply with the treatment assignment from a doctor, but some of them may comply, which is also called an indirect experiment in Pearl (1995).

If we perform an external intervention on V_i such that V_i has a conditional distribution $P'(v_i \mid pa(v_i))$ different from the pre-intervention distribution $P(v_i \mid pa(v_i))$ in (1) and other distributions are kept unchanged, then we have the post-intervention joint distribution

$$P_{V_i}(v_1, v_2, \cdots, v_n) = P'(v_i \mid pa(v_i)) \prod_{j \neq i} P(v_j \mid pa(v_j)).$$

In the external intervention, we may not be able to manipulate V_i, but we only need to change its conditional distribution, which may still depend on its parent variables. We call such an experiment a quasi-experiment. Below we discuss how to orient undirected edges via such quasi-experiments. Let τ be a chain component of the essential graph G^*, $ne(V_k)$ be the neighbor set of V_k, C be the children of V_k outside τ (that is, $C = ch(V_k) \setminus \tau$), and B be the set of all potential parents of V_k, that is, $B = ne(V_k) \setminus C$ is the neighbor set of V_k minus the children of V_k which have been identified in the chain graph. Let $V_i - V_k$ be an undirected edge in a chain component τ, and we want to orient the undirected edge by manipulating V_i. Since B is the neighbor set of V_k, we have $V_i \in B$ and thus $B \neq \emptyset$. Below we show a result which can be used to identify the direction of the undirected edge $V_i - V_k$ via a quasi-experiment of intervention on V_i.

Theorem 7 *For a quasi-experiment of intervention on V_i, we have the following properties*

1. $P_{V_i}(v_k \mid B) = P(v_k \mid B)$ *for all v_k and B if V_i is a parent of V_k, and*
2. $P_{V_i}(v_k) = P(v_k)$ *for all v_k if V_i is a child of V_k.*

According to Theorem 7, we can orient the undirected edge $V_i - V_k$ as

1. $V_i \leftarrow V_k$ if $P_{V_i}(v_k \mid B) \neq P(v_k \mid B)$ for some v_k and B, or
2. $V_i \rightarrow V_k$ if $P_{V_i}(v_k) \neq P(v_k)$ for some v_k.

The nonequivalence of pre- and post-intervention distributions is tested by using both experimental data and observational data, which is different from that of randomized experiments.

Example 1 (continued). Consider again the essential graph in Figure 3. We use a quasi-experiment of manipulating V_1 in order to orient the undirected edges connecting V_1 ($V_3 - V_1 - V_2$). We may test separately four null hypotheses $P_{V_1}(v_2) = P(v_2)$, $P_{V_1}(v_3) = P(v_3)$, $P_{V_1}(v_2 \mid v_1, v_3, v_4) = P(v_2 \mid v_1, v_3, v_4)$ and $P_{V_1}(v_3 \mid v_1, v_2, v_4) = P(v_3 \mid v_1, v_2, v_4)$ with both observational and experimental data. We orient $V_1 - V_2$ as $V_1 \rightarrow V_2$ if $P_{V_1}(v_2) \neq P(v_2)$, otherwise as $V_1 \leftarrow V_2$ (or further check whether there is a stronger evidence of $P_{V_1}(v_2 \mid v_1, v_3, v_4) \neq P(v_2 \mid v_1, v_3, v_4)$). Similarly we can orient $V_1 - V_3$. Finally we obtain four possible orientations as shown in Table 1.

If both $P_{V_i}(v_k) = P(v_k)$ and $P_{V_i}(v_k \mid B) = P(v_k \mid B)$ for all v_k and B hold for a quasi-experiment, then we cannot identify the direction of edge $V_i - V_k$ from the intervention. For example, suppose that there are only two variables V_1 and V_2, V_1 has three levels and V_1 is the parent of V_2. If the true conditional distribution of V_2 given V_1 is: $p(v_2 \mid V_1 = 1) = p(v_2 \mid V_1 = 2) \neq p(v_2 \mid V_1 = 3)$, then the undirected edge $V_1 - V_2$ cannot be oriented with the intervention on V_1 with $p_{V_1}(V_1 = v) \neq p(V_1 = v)$ for $v = 1$ and 2 but $p_{V_1}(V_1 = 3) = p(V_1 = 3)$ because we have that $p_{V_1}(v_2) = p(v_2)$ for all v_2 and that $p_{V_1}(v_2 \mid B) = p(v_2 \mid B)$ where $B = \{V_1\}$. In a quasi-experiment, an

experimenter may not be able to manipulate V_1, and thus this phenomenon can occur. If V_1 can be manipulated, then the experimenter can choose the distribution of V_2 to avoid this phenomenon.

4. Optimal Designs of Intervention Experiments

In this section, we discuss the optimal designs of intervention experiments which are used to minimize the number of manipulated variables or to minimize the uncertainty of candidate structures after an intervention experiment based on some criteria. Since the orientation for one chain component is unrelated to the orientations for other components, we can design an intervention experiment for each chain component separately. As shown in Section 2, given a chain component τ, we orient the subgraph over τ into a DAG G_τ without any v-structure or cycle via experiments of interventions in variables in τ. For simplicity, we omit the subscript τ in this section. In the following subsections, we discuss intervention designs for only one chain component. We first introduce the concept of sufficient interventions and discuss their properties of sufficient interventions, then we present the optimal design of batch interventions, and finally we give the optimal design of sequential interventions. For optimizing quasi-experiments of interventions, we assume that intervention on a variable V_i will change the marginal distribution of its child V_j, that is, there is a level v_j such that $P_{V_i}(v_j) \neq P(v_j)$ for $V_i \to V_j$. Under this assumption, all undirected edges connecting a node V_i can be oriented via a quasi-experiment of intervention on variable V_i. Without the assumption, there may be some undirected edge which cannot be oriented even if we perform interventions in both of its two nodes.

4.1. Sufficient Interventions

It is obvious that we can identify a DAG in a Markov equivalence class if we can manipulate all variables which connect undirected edges. However, it may be unnecessary to manipulate all of these variables. Let $\mathcal{S} = (V_1, V_2, \cdots, V_k)$ denote a sequence of manipulated variables. We say that a sequence of manipulated variables is sufficient for a Markov equivalence class $[G]$ if we can identify one DAG from all possible DAGs in $[G]$ after these variables in the sequence are manipulated. That is, we can orient all undirected edges of the essential graph G^* no matter which G in $[G]$ is the true DAG. There may be several sufficient sequences for a Markov equivalence class $[G]$.

Let g denote the number of nodes in the chain component, and h the number of undirected edges within the component. Then there are at most 2^h possible orientation of these undirected edges, and thus there are at most 2^h DAGs over the component in the Markov equivalence class. Given a permutation of nodes in the component, a DAG can be obtained by orienting all undirected edges backwards in the direction of the permutation, and thus there are at most $\min\{2^h, g!\}$ DAGs in the class.

Theorem 8 *If a sequence $\mathcal{S} = (V_1, V_2, \cdots, V_k)$ of manipulated variables is sufficient, then any permutation of $\mathcal{S}$ is also sufficient.*

According to Theorem 8, we can ignore the order of variables in an intervention sequence and treat the sequence as a variable set. Thus, if $\mathcal{S}$ is a sufficient set, then $\mathcal{S}'$ which contains $\mathcal{S}$ is also sufficient. Manipulating V_i, we obtain a class $E(V_i) = \{e(V_i)\}$ of orientations (see Table 1 as an example). Given an orientation $e(V_i)$, we can obtain the class $[G]_{e(V_i)}$ by (3). We say that $e(V_1, \ldots, V_k) = \{e(V_1), \ldots, e(V_k)\}$ is a legal combination

of orientations if there is not any v-structure or cycle formed and there is not any undirected edge oriented in two different directions by these orientations. For a set $\mathcal{S} = (V_1, \ldots, V_k)$ of manipulated variables, the Markov equivalence class is reduced into a class

$$[G]_{e(V_1,\ldots,V_k)} = [G]_{e(V_1)} \cap \ldots \cap [G]_{e(V_k)}$$

for a legal combination $e(V_1, \ldots, V_k)$ of orientations. If $[G]_{e(V_1,\ldots,V_k)}$ has only one DAG for all possible legal combinations $e(V_1, \ldots, V_k) \in E(V_1) \times \ldots \times E(V_k)$, then the set $\mathcal{S}$ is a sufficient set for identifying any DAG in $[G]$. Let $\mathbb{S}$ denote the class of all sufficient sets, that is, $\mathbb{S} = \{\mathcal{S} : \mathcal{S} \text{ is sufficient}\}$. We say that a sequence $\mathcal{S}$ is minimum if any subset of $\mathcal{S}$ is not sufficient.

Theorem 9 *The intersection of all sufficient sets is an empty set, that is, $\bigcap_{\mathcal{S}\in\mathbb{S}} \mathcal{S} = \varnothing$. In addition, the intersection of all minimum sufficient sets is also an empty set.*

From Theorem 9, we can see that there is not any variable that must be manipulated to identify a causal structure. Especially, any undirected edge can be oriented by manipulating either of its two nodes.

4.2. Optimization for Batch Interventions

We say that an intervention experiment is a batch-intervention experiment if all variables in a sufficient set $\mathcal{S}$ are manipulated in a batch to orient all undirected edges of an essential graph. Let $|\mathcal{S}|$ denote the number of variables in $\mathcal{S}$. We say that a batch intervention design is optimal if its sufficient set $\mathcal{S}_o$ has the smallest number of manipulated variables, that is, $|\mathcal{S}_o| = \min\{|\mathcal{S}| : \mathcal{S} \in \mathbb{S}\}$. Given a Markov equivalence class $[G]$, we try to find a sufficient set S which has the smallest number of manipulated variables for identifying all possible DAGs in the class $[G]$. Below we give an algorithm to find the optimal design for batch interventions, in which we first try all sets with a single manipulated variable, then try all sets with two variables, and so on, until each post-intervention Markov equivalence class has a single DAG.

Given a Markov equivalence class $[G]$, we manipulate a node V and obtain an orientation of some edges, denoted by $e(V)$. The class $[G]$ is split into several subclasses, denoted by $[G]_{e(V)}$ for all possible orientations $e(V)$. Let $[G]_{e(V_1,V_2)}$ denote a subclass with an orientation obtained by manipulating V_1 and V_2. The following algorithm 1 performs exhaustive search for the optimal design of batch interventions. Before calling Algorithm 1, we need to enumerate all DAGs in the class $[G]$, and then we can easily find $[G]_{e(V_i)}$ according to (3). There are at most $\min\{g!, 2^h\}$ DAGs in the class $[G]$, and thus the upper bound of the complexity for enumerating all $\{[G]_{e(V_i)}\}$ is $\min\{g!, 2^h\}$. We may be able to have an efficient method to find all $\{[G]_{e(V_i)}\}$ using the structure of the chain component.

Algorithm 1 exhaustively searches all combinations of manipulated variables to find the minimum sufficient sets, and its complexity is $O(g!)$, although Algorithm 1 may stop whenever it finds some minimum sets. The calculations in Algorithm 1 are only simple set operations

$$[G]_{e(\mathcal{S})} = [G]_{e(V_{i_1})} \cap \ldots \cap [G]_{e(V_{i_k})},$$

where all $[G]_{e(V_i)}$ have been found before calling Algorithm 1. Notice that a single chain component usually has a size g much less than the total number n of variables. Algorithm 1 is feasible for a mild size g. A more efficient algorithm or a greedy method

Table 2: The intervention process to identify a causal structure from the essential graph in Figure 3, where $*$ means that the intervention is unnecessary.

V_1	V_2	V_3	V_4	DAG in Fig. 4
$V_1 \perp\!\!\!\perp V_2$ *and* $V_1 \not\perp\!\!\!\perp V_3$	*	*	*	(3)
$V_1 \not\perp\!\!\!\perp V_2$ *and* $V_1 \perp\!\!\!\perp V_3$	*	*	*	(6)
$V_1 \not\perp\!\!\!\perp V_2$ *and* $V_1 \not\perp\!\!\!\perp V_3$	$V_2 \perp\!\!\!\perp V_3$	*	*	(1)
	$V_2 \not\perp\!\!\!\perp V_3$	*	*	(2)
$V_1 \perp\!\!\!\perp V_2$ *and* $V_1 \perp\!\!\!\perp V_3$	$V_2 \perp\!\!\!\perp V_3$ *and* $V_2 \not\perp\!\!\!\perp V_4$	*	*	(7)
	$V_2 \not\perp\!\!\!\perp V_3$ *and* $V_2 \not\perp\!\!\!\perp V_4$	$V_3 \not\perp\!\!\!\perp V_4$	*	(4)
		$V_3 \perp\!\!\!\perp V_4$	*	(5)
	$V_2 \perp\!\!\!\perp V_3$ *and* $V_2 \perp\!\!\!\perp V_4$	$V_3 \not\perp\!\!\!\perp V_4$	*	(8)
		$V_3 \perp\!\!\!\perp V_4$	$V_4 \not\perp\!\!\!\perp V_5$	(9)
			$V_4 \perp\!\!\!\perp V_5$	(11)
	$V_2 \not\perp\!\!\!\perp V_3$ *and* $V_2 \perp\!\!\!\perp V_4$	*	$V_4 \not\perp\!\!\!\perp V_5$	(10)
			$V_4 \perp\!\!\!\perp V_5$	(12)

Algorithm 1: Algorithm for finding the optimal designs of batch interventions

Input: A chain graph G induced by a chain component $\tau = \{V_1, \ldots, V_g\}$, and $[G]_{e(V_i)}$ for all $e(V_i)$ and i.

Output: All optimal designs of batch interventions.

Initialize the size k of the minimum intervention set as $k = 0$.
repeat
  Set $k = k + 1$.
  for all possible variable subsets $\mathcal{S} = \{V_{i_1}, \ldots, V_{i_k}\}$ **do**
    if $|[G]_{e(\mathcal{S})}| = 1$ for all possible legal combination $e(\mathcal{S})$ of orientations **then**
      return the minimum sufficient set $\mathcal{S}$
    end if
  end for
until find some sufficient sets

is needed for a large g and h. In this case, there are too many DAGs to enumerate. We can first take a random sample of DAGs from the class $[G]$ with the simulation method proposed in the next subsection, and then we use the sample approximately to find an optimal design.

A possible greedy approach is to select a node to be first manipulated from the chain component which has the largest number of neighbors such that the largest number of undirected edges are oriented by manipulating it, and then delete these oriented edges. Repeat this process until there is not any undirected edge left. But there are cases where the sufficient set obtained from the greedy method is not minimum.

Example 1 (continued). Consider the essential graph in Figure 3, which depicts a Markov equivalence class with 12 DAGs in Figure 4. From Algorithm 1, we can find that $\{1,2,4\}$, $\{1,3,4\}$, $\{2,3,4\}$ and $\{2,3,5\}$ are all the minimum sufficient sets. The greedy method can obtain the same minimum sufficient sets for this example.

4.3. Optimization for Sequential Interventions

The optimal design of batch interventions presented in the previous subsection tries to find a minimum sufficient set $\mathcal{S}$ before any variable is manipulated, and thus it cannot use orientation results obtained by manipulating the previous variables during the intervention process. In this subsection, we propose an experiment of sequential interventions, in which variables are manipulated sequentially. Let $\mathcal{S}^{(t)}$ denote the set of variables that have been manipulated before step t and $\mathcal{S}^{(0)} = \emptyset$. At step t of the sequential experiment, according to the current Markov equivalence class $[G]_{e(\mathcal{S}^{(t-1)})}$ obtained by manipulating the previous variables in $\mathcal{S}^{(t-1)}$, we choose a variable V to be manipulated based on some criterion. We consider two criteria for choosing a variable. One is the minimax criterion based on which we choose a variable V such that the maximum size of subclasses $[G]_{e(\mathcal{S}^{(t)})}$ for all possible orientations $e(\mathcal{S}^{(t)})$ is minimized. The other is the maximum entropy criterion based on which we choose a variable V such that the following entropy is maximized for any V in the chain component τ

$$H_V = -\sum_{i=1}^{M} \frac{l_i}{L} \log \frac{l_i}{L},$$

where l_i denotes the number of possible DAGs of the chain component with the ith orientation $e(V)_i$ obtained by manipulating V, $L = \sum_i l_i$ and M is the number of all possible orientations $e(V)_1, \ldots, e(V)_M$ obtained by manipulating V. Based on the maximum entropy criterion, the post-intervention subclasses have sizes as small as possible and they have sizes as equal as possible, which means uncertainty for identifying a causal DAG from the Markov equivalence class is minimized by manipulating V. Below we give two examples to illustrate how to choose variables to be manipulated in the optimal design of sequential interventions based on the two criteria.

Example 1 (continued). Consider again the essential graph in Figure 3, which depicts a Markov equivalence class with 12 DAGs in Figure 4. Tables 3 to 6 show the results for manipulating one of variables V_1, V_2 (symmetry to V_3), V_4 and V_5 respectively in order to distinguish the possible DAGs in Figure 4. The first row in these tables gives possible orientations obtained by manipulating the corresponding variable. The second row gives DAGs obtained by the orientation, where numbers are used to index DAGs in Figure 4. The third row gives the number l_i of DAGs of this chain component for the ith orientation. The entropies for manipulating $V_1, \ldots, V_5$ are 0.9831, 1.7046, 1.7046, 1.3480,

0.4506, respectively. Based on the maximum entropy criterion, we choose variable V_2 or V_3 to be manipulated first. The maximum numbers l_i of DAGs for manipulating one of $V_1, \ldots, V_5$ are 8, 3, 3, 6, 10, respectively. Based on the minimax criterion, we also choose variable V_2 or V_3 to be manipulated first.

Table 3: Manipulating V_1

Orientation	$V_2 \leftarrow V_1 \rightarrow V_3$	$V_2 \rightarrow V_1 \rightarrow V_3$	$V_2 \rightarrow V_1 \leftarrow V_3$	$V_2 \leftarrow V_1 \leftarrow V_3$
DAGs	$\{1,2\}$	$\{3\}$	$\{4,5,7,8,9,10,11,12\}$	$\{6\}$
l_i	2	1	8	1
Entropy is 0.9831 and maximum l_i is 8				

Table 4: Manipulating V_2

Orientation						
DAGs	$\{8,9,11\}$	$\{10,12\}$	$\{3,4,5\}$	$\{2\}$	$\{1,6\}$	$\{7\}$
l_i	3	2	3	1	2	1
Entropy is 1.7046 and maximum l_i is 3						

Table 5: Manipulating V_4

Orientation					
DAGs	$\{1,2,3,4,6,7\}$	$\{5\}$	$\{8\}$	$\{9,10\}$	$\{11,12\}$
l_i	6	1	1	2	2
Entropy is 1.3480 and maximum l_i is 6					

Although the same variable V_2 or V_3 is chosen to be manipulated first in the above example, in general, the choice may be different based on the two criteria. The minimax criterion tends to be more conservative, and the entropy criterion tends to be more uniform. For example, consider two interventions for an equivalence class with 10 DAGs: one splits the class into 8 subclasses with the numbers $(l_1, \ldots, l_8) = (1, 1, 1, 1, 1, 1, 1, 3)$ of DAGs, the other splits it into 5 subclasses with the numbers of DAGs equal to (2,2,2,2,2). Then the minimax criterion chooses the second intervention, while the maximum entropy criterion chooses the first intervention.

To find the number (l_i for $i = 1, \cdots, M$), we need to enumerate all DAGs in the class $[G]$ and then count the number l_i of DAGs with the same orientations as $e(V)_i$. As discussed in Section 4.2, the upper bound of the complexity for calculating all l_i is $O(\min\{g!, 2^h\})$. Generally the size g of a chain component is much less than the number n of the full variable set and the number h of undirected edges in a chain component is

Table 6: Manipulating V_5

Orientation	$V_4 \to V_5$	$V_4 \leftarrow V_5$
DAGs	$\{1,2,3,4,5,6,7,8,9,10\}$	$\{11,12\}$
l_i	10	2
Entropy is 0.4506 and maximum l_i is 10		

not very large. In the following example, we show a special case with a tree structure, where the calculation is easy.

Example 2 In this example, we consider a special case that a chain component has a tree structure. It does not mean that a DAG is a tree, and it is not uncommon in a chain component (see Figure 1). Since there are no v-structures in any chain component, all undirected edges in a subtree can be oriented as long as we find its root. Manipulating a node V in a tree, we can determinate all orientations of edges connecting V, and thus all subtrees that are emitted from V can be oriented, but only one subtree with V as a terminal cannot be oriented. Suppose that node V connects M undirected edges, and let l_i denote the number of nodes in the ith subtree connecting V for $i = 1, \ldots, M$. Since each node in the ith subtree may be the root of this subtree, there are l_i possible orientations for the ith subtree. Thus we have the entropy for manipulating V

$$H_V = -\sum_{i=1}^{M} \frac{l_i}{L} log \frac{l_i}{L}.$$

Consider the chain component $\tau = \{V_1, \ldots, V_4\}$ of the chain graph G^* in Figure 1, which has a tree structure. In Table 7, the first column gives variables to be manipulated, the second column gives possible orientations via the intervention, the third column gives the equivalence subclasses (see Figure 2) for each orientation, the fourth column gives the number l_i of possible DAGs for the ith orientation and the last column gives the entropy for each intervention. From Table 7, we can see that manipulating V_1 or V_2 has the maximum entropy and the minimax size.

Table 7: Manipulating variables in a chain component with a tree structure.

Intervention	Orientation	Subclass of DAGs	l_i	H_V
V_1	$V_2 \leftarrow V_1 \to V_3$	G	1	1.0397
	$V_2 \to V_1 \to V_3$	G_1, G_2	2	
	$V_2 \leftarrow V_1 \leftarrow V_3$	G_3	1	
V_2	$V_4 \leftarrow V_2 \leftarrow V_1$	G, G_3	2	1.0397
	$V_4 \leftarrow V_2 \to V_1$	G_1	1	
	$V_4 \to V_2 \to V_1$	G_2	1	
V_3	$V_1 \to V_3$	G, G_1, G_2	3	0.5623
	$V_1 \leftarrow V_3$	G_3	1	
V_4	$V_4 \leftarrow V_2$	G, G_1, G_3	3	0.5623
	$V_4 \to V_2$	G_2	1	

An efficient algorithm or an approximate algorithm is necessary when both g and h are very large. A simulation algorithm can be used to estimate l_i/L. In this simulation

method, we randomly take a sample of DAGs without any v-structure from the class $[G]$. To draw such a DAG, we randomly generate a permutation of all nodes in the class, orient all edges backwards in the direction of the permutation, and keep only the DAG without any v-structure. There may be some DAGs in the sample which are the same, and we keep only one of them. Then we count the number l_i' of DAGs in the sample which have the same orientation as $e(V)_i$. We can use l_i'/L' to estimate l_i/L, where $L' = \sum_i l_i'$. When the sample size tends to infinite, all DAGs in the class can be drawn, and then the estimate l_i'/L' tends to l_i/L. Another way to draw a DAG is that we randomly orient each undirected edge of the essential graph, but we need to check whether there is any cycle besides v-structure.

5. Simulation

In this section, we use two experiments to evaluate the active learning approach and the optimal designs via simulations. In the first experiment, we evaluate a whole process of structural learning and orientation in which we first find an essential graph using the PC algorithm and then orient the undirected edges using the approaches proposed in this paper. In the second experiment, we compare various designs for orientations starting with the same underlying essential graph. For both experiments, the DAG (1) in Figure 4 is used as the underlying DAG and all variables are binary. Its essential graph is given in Figure 3 and there are other 11 DAGs which are Markov equivalent to the underlying DAG (1), as shown in Figure 4. This essential graph can also be seen as a chain component of a large essential graph. All conditional probabilities $P(v_j \mid pa(v_j))$ are generated from the uniform distribution $U(0,1)$. We repeat 1000 simulations with the sample size $n = 1000$.

In each simulation of the first experiment, we first use the PC algorithm to find an essential graph with the significance level $\alpha = 0.15$ with which the most number of true essential graphs were obtained among various significance levels in our simulations. Then we use the intervention approach proposed in Section 3 to orient undirected edges of the essential graph. To compare the performances of orientations for different significance levels and sample sizes used in intervention experiments, we run simulations for various combinations of significance levels $\alpha_I = 0.01, 0.05, 0.10, 0.15, 0.20, 0.30$ and sample sizes $n_I = 50, 100, 200, 500$ in intervention experiments. To compare the performance of the experiment designs, we further give the numbers of manipulated variables that are necessary to orient all undirected edges of the same essential graphs in various intervention designs. We run the simulations using R 2.6.0 on an Intel(R) Pentium(R) M Processor with 2.0 GHz and 512MB RAM and MS XP. It takes averagely 0.4 second of the processor time for a simulation, and each simulation needs to finish the following works: (1) generate a joint distribution and then generate a random sample of size $n = 1000$, (2) find an essential graph using the PC algorithm, (3) find an optimal design, and (4) repeatedly generate experimental data of size n_I until identifying a DAG.

To make the post-intervention distribution $P'(v_i \mid pa(v_i))$ different from the pre-intervention $P(v_i \mid pa(v_i))$, we use the post-intervention distribution of the manipulated variable V_i as follows

$$P'(v_i \mid pa(v_i)) = P'(v_i) = \begin{cases} 1, & P(v_i) \leq 0.5; \\ 0, & \text{otherwise.} \end{cases}$$

To orient an undirected edge $V_i - V_j$, we implemented both the independence test of the manipulated V_i and its each neighbor variable V_j for randomized experiments and the equivalence test of pre- and post-intervention distributions (i.e., $P_{V_i}(v_j) = P(v_j)$ for all v_j) in our simulations. Both tests have the similar results and the independence test is little more efficient than the equivalence test. To save space, we only show the simulation results of orientations obtained by the equivalence test and the optimal design based on the maximum entropy criterion in Table 8, and other designs have the similar results of orientations.

To evaluate the performance of orientation, we define the percentage of correct orientations as the ratio of the number of correctly oriented edges to the number of edges that are obtained from the PC algorithm and belong to the DAG (1) in Figure 4. The third column λ in Table 8 shows the average percentages of correctly oriented edges of the DAG (1) in 1000 simulations. To separate the false orientations due to the PC algorithm from those due to intervention experiments, we further check the cases that the essential graph in Figure 3 is correctly obtained from the PC algorithm. The fourth column m shows the number of correct essential graphs obtained from the PC algorithm in 1000 simulations. In the fifth column, we show the percentage λ' of correct orientations for the correct essential graph. Both λ and λ' increase as n_I increases. Comparing λ and λ', it can be seen that there are more edges oriented correctly when the essential graph is correctly obtained from the PC algorithm. From the sixth to eleven columns, we give the cumulative distributions of the number of edges oriented correctly when the essential graph is correctly obtained. The column labeled '$\geq i$' means that we correctly oriented more than or equal to i of 6 edges of the essential graph in Figure 3, and the values in this column denote the percents of DAGs with more than or equal to i edges correctly oriented in those simulations. For example, the column '≥ 5' means that more than or equal to 5 edges are oriented correctly (i.e., the DAGs (1), (2) and (6) in Figure 4), and 0.511 in the first line means that 51.1% of $m = 409$ correct essential graphs were oriented with '≥ 5' correct edges. The column '6' means that the underlying DAG (1) is obtained correctly. From this column, it can be seen that more and more DAGs are identified correctly as the size n_I increases. The cumulative distribution for ≥ 0 is equal to one and is omitted. From these columns, it can be seen that more and more edges are correctly oriented as the size n_I increases. From λ and λ', we can see that a larger α_I is preferable for a smaller size n_I, and a smaller α_I is preferable for a larger n_I. For example, $\alpha_I = 0.20$ is the best for $n_I = 50$, $\alpha_I = 0.10$ for $n_I = 100$, $\alpha_I = 0.05$ for $n_I = 200$, $\alpha_I = 0.01$ for $n_I = 500$.

In the second experiment, we compare the numbers of manipulated variables to orient the same underlying essential graph for different experimental designs. In the following simulations, we set $n_I = 100$ and $\alpha_I = 0.1$, and all orientations start with the true essential graph in Figure 3. As shown in Section 4.2, the optimal batch design and the design by the greedy method always need three variables to be manipulated for orientation of the essential graph. For the optimal sequential designs, the frequencies of the numbers of manipulated variables in 1000 simulations are given in Table 9. In the random design labeled 'Random', we randomly select a variable to be manipulated at each sequential step, only one variable is manipulated for orientations in 268 of 1000 simulations, and four variables are manipulated in 55 of 1000 simulations. In the middle of Table 9, we show the simulation results of the optimal sequential designs based on the minimax criterion and its approximate designs obtained by drawing a sample of DAGs. The minimax design needs only one or two variables to be manipulated in all 1000 simulations. We show three approximate designs which draw h, $h \times 5$ and $h \times 10$

Table 8: The simulation results

					The number of edges oriented correctly					
n_I	α_I	λ	m	λ'	6	≥ 5	≥ 4	≥ 3	≥ 2	≥ 1
50	.01	.672	409	.758	0.401	0.511	0.868	0.870	0.927	0.973
	.05	.699	409	.782	0.496	0.616	0.829	0.839	0.934	0.976
	.10	.735	418	.808	0.538	0.646	0.833	0.868	0.969	0.993
	.15	.745	407	.821	0.516	0.690	0.855	0.909	0.966	0.990
	.20	.756	404	.826	0.564	0.723	0.832	0.899	0.963	0.978
	.30	.741	373	.819	0.501	0.729	0.823	0.920	0.965	0.979
100	.01	.761	401	.850	0.586	0.706	0.910	0.925	0.975	0.995
	.05	.774	408	.846	0.588	0.721	0.885	0.919	0.973	0.993
	.10	.806	425	.878	0.668	0.814	0.896	0.925	0.974	0.993
	.15	.794	410	.868	0.624	0.790	0.878	0.932	0.985	1.000
	.20	.788	382	.875	0.626	0.812	0.890	0.948	0.982	0.992
	.30	.798	417	.861	0.583	0.777	0.856	0.959	0.988	1.000
200	.01	.822	421	.901	0.724	0.808	0.945	0.948	0.988	0.995
	.05	.836	402	.911	0.701	0.853	0.950	0.973	0.995	0.995
	.10	.833	408	.900	0.686	0.863	0.917	0.949	0.993	0.995
	.15	.823	382	.901	0.696	0.851	0.911	0.955	0.995	1.000
	.20	.826	395	.886	0.658	0.820	0.889	0.962	0.990	0.997
	.30	.822	402	.887	0.614	0.828	0.905	0.975	0.998	1.000
500	.01	.870	369	.966	0.878	0.943	0.984	0.992	1.000	1.000
	.05	.869	388	.940	0.802	0.920	0.951	0.977	0.995	0.997
	.10	.863	399	.936	0.762	0.905	0.952	0.995	1.000	1.000
	.15	.859	433	.926	0.723	0.898	0.956	0.986	0.995	1.000
	.20	.846	390	.923	0.703	0.890	0.956	0.990	0.997	1.000
	.30	.834	389	.893	0.599	0.820	0.949	0.992	1.000	1.000

DAGs from a chain component with h undirected edges respectively. For example, the sample sizes of DAGs from the initial essential graph $[G]$ with $h = 6$ undirected edges are 6, 30 and 60, respectively. As the sample size increases, the distribution of the manipulated variable numbers tends to the distribution for the exact minimax design. The optimal sequential design based on the maximum entropy criterion has a very similar performance as that based on the minimax criterion, as shown in the bottom of Table 9. According to Table 9, all of the sequential intervention designs (Random, Minimax, Entropy and their approximations) are more efficient than the batch design, and the optimal designs based on the minimax and the maximum entropy criteria are more efficient than the random design.

Table 9: The frequencies of the numbers of interventions

	m^*			
Design	1	2	3	4
Random	268	475	202	55
Minimax	437	563	0	0
Approx. (h)	372	469	159	0
Approx. ($h \times 5$)	413	573	14	0
Approx. ($h \times 10$)	426	574	0	0
Entropy	441	559	0	0
Approx. (h)	375	454	171	0
Approx. ($h \times 5$)	435	547	18	0
Approx. ($h \times 10$)	425	574	1	0

m^* denotes the number of manipulated variables

6. Conclusions

In this paper, we proposed a framework for active learning of causal structures via intervention experiments, and further we proposed optimal designs of batch and sequential interventions based on the minimax and the maximum entropy criteria. A Markov equivalence class can be split into subclasses by manipulating a variable, and a causal structure can be identified by manipulating variables repeatedly. We discussed two kinds of external intervention experiments, the randomized experiment and the quasi-experiment. In a randomized experiment, the distribution of a manipulated variable does not depend on its parent variables, while in a quasi-experiment, it may depend on its parents. For a randomized experiment, the orientations of an undirected edge can be determined by testing the independence of the manipulated variable and its neighbor variable only with experimental data. For a quasi-experiment, the orientations can be determined by testing the equivalence of pre- and post-intervention distributions with both experimental and observational data. We discussed two optimal designs of batch and sequential interventions. For the optimal batch design, a smallest set of variables to be manipulated is found before interventions, which is sufficient to orient all undirected edges of an essential graph. But the optimal batch design does not use orientation results obtained by manipulating the previous variables during the intervention process, and thus it may be less efficient than the optimal sequential designs. For the optimal sequential design, we choose a variable to be manipulated sequentially such that the

current Markov equivalence class can be reduced to a subclass with potential causal DAGs as little as possible. We discussed two criteria for optimal sequential designs, the minimax and the maximum entropy criteria. The exact, approximate and greedy methods are presented for finding the optimal designs.

The scalability of the optimal designs proposed in this paper depends only on the sizes of chain components but does not depend on the size of a DAG since the optimal designs are performed separately within every chain component. As discussed in Section 4, the optimal designs need to find the number of possible DAGs in a chain component, which has a upper bound $\min\{2^h, g!\}$. When both the number h of undirected edges and the number g of nodes in a chain component are very large, instead of using the optimal designs, we may use the approximate designs via sampling DAGs. We checked several standard graphs found at the Bayesian Network Repository (http://compbio.cs.huji.ac.il/Repository/). We extracted their chain components and found that most of their chain components have tree structures and their sizes are not large. For example, ALARM with 37 nodes has 4 chain components with only two nodes in each component, HailFinder with 56 nodes has only one component with 18 nodes, Carpo with 60 nodes has 9 components with at most 7 nodes in each component, Diabets with 413 nodes has 25 components with at most 3 nodes, and Mumin 2 to Mumin 4 with over 1000 nodes have at most 21 components with at most 35 nodes. Moreover, all of those largest chain components have tree structures, and thus we can easily carry out optimal designs as discussed in Example 2.

In this paper, we assume that there are no latent variables. Though the algorithm can orient the edges of an essential graph and output a DAG based on a set of either batch or sequential interventions, the application of the method for learning causality in the real word is pretty limited because latent or hidden variables are typically present in real-world data sets.

Acknowledgments

We would like to thank the guest editors and the three referees for their helpful comments and suggestions that greatly improved the previous version of this paper. This research was supported by Doctoral Program of Higher Education of China (20070001039), NSFC (70571003, 10771007, 10431010), NBRP 2003CB715900, 863 Project of China 2007AA01Z43, 973 Project of China 2007CB814905 and MSRA.

Appendix A. Proofs of Theorems

Before proving Theorems 4 and 5, we first give a lemma which will be used in their proofs.

Lemma 10 *If a node $V \in \mathbb{V}$ is a parent of a node U in a chain component τ of G^* (i.e., $(V \to U) \in G^*$, $U \in \tau$, $V \in \mathbb{V}$ and $V \notin \tau$), then V is a parent of all nodes in τ (i.e., $(V \to W) \in G$ for any $W \in \tau$).*

Proof By (iii) of Lemma 3, $V \to U—W$ does not occur in any induced subgraph of G^*. Thus for any neighbor of U in the chain component τ, W and V must be adjacent in G^*. Because $V \notin \tau$, the edge between V and W is directed. There are two alternatives as shown in Figures 5 and 6 for the subgraph induced by $\{V, U, W\}$.

If it is the subgraph in Figure 6 (i.e., the $V \to W \in G'$ for any $G' \in [G]$), then $W \to U$ must be in G' for any $G' \in [G]$ in order to avoid a directed cycle, as shown in

Figure 7. So $W \to U$ must be in G^*. It is contrary to the fact that $\{U, W\} \in \tau$ is in a chain component of G^*. So V must also be a parent of W. Because all variables in τ are connected by undirected edges in G^*_τ, V must be a parent of all other variables in τ. ■

Proof of Theorem 4. According to Lemma 10, if a node W outside a component τ points at a node V in τ, then W must point at each node U in τ. Thus W, V and U cannot form a v-structure. ■

Proof of Theorem 5. Suppose that Theorem 5 does not hold, that is, there is a directed path $V_1 \to \cdots \to V_k$ in G_τ which is not a directed cycle, but $W_1 \to \cdots \to W_i \to V_1 \to \cdots \to V_k \to W_{i+1} \to \cdots \to W_1$ is a directed cycle, where $W_i \notin \tau$. We denote this cycle as DC. From Lemma 10, W_i must also be a parent of V_k, and thus $W_1 \to \cdots \to W_i \to V_k \to W_{i+1} \to \cdots \to W_1$ is also a directed cycle, denoted as DC'. Now, every edge of DC' is out of G_τ. Similarly, we can remove all edges in other chain components from DC' and keep the path being a directed cycle. Finally, we can get a directed cycle in the directed subgraph of G^*. It contradicts the fact that G^* is an essential graph of a DAG. So we proved Theorem 5. ■

To prove Theorem 6, we first present an algorithm for finding the post-intervention essential graph $G^*_{e(V)}$ via the orientation $e(V)$, then we show the correctness of the algorithm using several lemmas, and finally we give the proof of Theorem 6 with $G^*_{e(V)}$ obtained by the algorithm. In order to prove that $G^*_{e(V)}$ is also a chain graph, we introduce an algorithm (similar to Step D of SGS and the PC algorithm in Spirtes et al. (2000)) for constructing a graph, in which some undirected edges of the initial essential graph are oriented with the information of $e(V)$. Let τ be a chain graph of G^*, $V \in \tau$ and $e(V)$ be an orientation of undirected edges connecting V.

Algorithm 2: Find the post-intervention essential graph via orientation $e(V)$
Input: The essential graph G^* and $e(V)$
Output: The graph H
Orient the undirected edges connecting V in the essential graph G^* according to $e(V)$ and denote the graph as H.
Repeat the following two rules to orient some other undirected edges until no rules can be applied:
(i) if $V_1 \to V_2 - V_3 \in H$ and V_1 and V_3 are not adjacent in H, then orient $V_2 - V_3$ as $V_2 \to V_3$ and update H;
(ii) if $V_1 \to V_2 \to V_3 \in H$ and $V_1 - V_3 \in H$, then orient $V_1 - V_3$ as $V_1 \to V_3$ and update H.
return the graph H

It can be shown that H constructed by Algorithm 2 is a chain graph and H is equal to the post-intervention essential graph $G^*_{e(V)}$. We show those results with the following three Lemmas.

Lemma 11 *Let G^* be the essential graph of DAG G, τ be a chain component of G^* and I be a DAG over τ. Then there is a DAG $G' \in [G]$ such that $I = G'_\tau$ if and only if I is a DAG with the same skeleton as G^*_τ and without v-structures.*

Proof If there is a DAG $G' \in [G]$ such that $I = G'_\tau$, we have from Lemma 1 that I is a DAG with the same skeleton as G^*_τ and without v-structures.

Let I be a DAG with the same skeleton as G^*_τ and without v-structures, and G' be any DAG in the equivalence class $[G]$. We construct a new DAG I' from G' by substituting the subgraph G'_τ of G' with I. I' has the same skeleton as G'. From Theorems 4 and 5, I' has the same v-structures as G'. Thus I' is equivalent to G' and $I' \in [G]$. ■

Lemma 12 *Let H be a graph constructed by Algorithm 2. Then H is a chain graph.*

Proof If H is not a chain graph, there must be a directed cycle in subgraph H_τ for some chain component of G^*. Moreover, G^*_τ is chordal and $H \subset G^*$, and thus H_τ is chordal too. So we can get a three-edge directed cycle in H_τ as given in Figure 8 or 9.

If Figure 9 is a subgraph of H obtained at some step of Algorithm 2, then the undirected edge b—c is oriented as $b \leftarrow c$ according to Algorithm 2. Thus only Figure 8 can be a subgraph of H.

According to Lemma 10, we have that the directed edge $d \rightarrow b$ is not in G^*. Since all edges connecting a have been oriented in Step 1 of Algorithm 2, $d \rightarrow b$ is not an edge connecting a. So $d \rightarrow b$ must be identified at step 2 of Algorithm 2. There are two situations, one is to avoid a v-structure as shown in Figure 10, the other is to avoid a directed cycle as Figure 13.

We can arrange all directed edges in H_τ in order of orientations performed at Step 2 of Algorithm 2. First, we prove that the directed edge $d \rightarrow b$ in Figure 8 is not the first edge oriented at Step 2 of Algorithm 2.

In the first case as Figure 10, if $d \rightarrow b$ is the first edge oriented at Step 2 of Algorithm 2, we have $d_1 = a$. Because b and a are not adjacent, and d—c is an undirected edge in H, we have that $d_1 \rightarrow c$ must be in H as Figure 11, where $d_1 = a$. Now we consider the subgraph b—$c \leftarrow d_1$. According to the rules (i) and (ii) in Algorithm 2, we have that $b \leftarrow c$ is in $G^*_{e(a)}$ as Figure 12, which contradicts the assumption that b—$c \in H$.

In the second case as Figure 13, if $d \rightarrow b$ is the first edge oriented at Step 2 of Algorithm 2, we have $d_1 = a$.

Considering the structure $d_1 \rightarrow b$—c and that d—c is an undirected edge in H, we have that $d_1 \rightarrow c$ must be in H as Figure 14. Now we consider the subgraph of $\{d, d_1, c\}$. By Algorithm 2, $d \rightarrow c$ is in H as Figure 15, which contradicts the assumption that d—$c \in H$. Thus we have that the first edge oriented at Step 2 of Algorithm 2 is not in any directed cycle. Suppose that the first k oriented edges at Step 2 of Algorithm 2 are not in any directed cycle. Then we want to prove that the $(k+1)th$ oriented edge is also not in a directed cycle.

Let $d \rightarrow b$ be the $(k+1)th$ oriented edge at Step 2 of Algorithm 2, and Figure 8 be a subgraph of H. There are also two cases as Figures 10 and 13 for orienting $d \rightarrow b$.

In the case of Figure 10, since $d_1 \rightarrow d$ is in the first k oriented edges and d—$c \in H$, we have that $d_1 \rightarrow c$ must be in H. We also get that $b \leftarrow c$ must be in H as Figure 12, which contradicts the assumption that b—$c \in H$.

In the case of Figure 10, since $d_1 \rightarrow b$ and $d \rightarrow d_1$ are in the first k oriented edges and b—$c \in H$, we have that $d_1 \rightarrow c$ must be in H. We also get that $d \leftarrow c$ must be in H as Figure 15, which contradicts the assumption that d—$c \in H$. So the $(k+1)$th oriented edge is also not in any directed cycle. Now we can get that every directed edge in H_τ is not in any directed cycle. It implies that there are no directed cycles in H_τ, and thus H is a chain graph. ■

Lemma 13 *Let $G^*_{e(V)}$ be the post intervention essential graph with the orientation $e(V)$ and H be the graph constructed by Algorithm 2. We have $G^*_{e(V)} = H$.*

Proof We first prove $G^*_{e(a)} \subseteq H$. We just need to prove that all directed edges in H must be in $G^*_{e(a)}$. We use induction to finish the proof.

After Step 1 of Algorithm 2, all directed edges in H are in $G^*_{e(a)}$. We now prove that the first directed edge oriented at Step 2 of Algorithm 2, such as $b \leftarrow c$, is in $G^*_{e(a)}$. Because $b \leftarrow c$ must be oriented by the rule (i) of Algorithm 2, there must be a node $d \notin \tau$ such that $b—c \leftarrow d$ is the subgraph of H. So $b \leftarrow c \leftarrow d$ must be a subgraph in each $G' \in G^*_{e(a)}$. Otherwise, $b \rightarrow c \leftarrow d$ forms a v-structure such that $G' \notin [G]$. Thus we have $b \leftarrow c \in G^*_{e(a)}$.

Suppose that the first k oriented edges at Step 2 of Algorithm 2 are in $G^*_{e(a)}$. We now prove that the $(k+1)$th oriented edge at Step 2 of Algorithm 2 is also in $G^*_{e(a)}$. Denoting the $(k+1)$th oriented edge as $l \leftarrow h$, according to the rules in Algorithm 2, there are two cases to orient $l \leftarrow h$ as shown in Figures 16 and 17.

In Figure 16, because $f \rightarrow h$ is in every DAG $G' \in G^*_{e(a)}$, in order to avoid a new v-structure, we have that $l \leftarrow h$ must be in every DAG $G' \in G^*_{e(a)}$. Thus we have $l \leftarrow h \in G^*_{e(a)}$. In Figure 17, because $l \rightarrow f$ and $f \rightarrow h$ are in every DAG $G' \in G^*_{e(a)}$, in order to avoid a directed cycle, we have that $h \leftarrow l$ must be in every DAG $G' \in G^*_{e(a)}$. Thus we have $h \leftarrow l \in G^*_{e(a)}$. Now we get that the $(k+1)$th oriented edge at Step 2 of Algorithm 2 is also in $G^*_{e(a)}$. Thus all directed edges in H are also in $G^*_{e(a)}$ and then we have $G^*_{e(a)} \subseteq H$.

Because H is a chain graph by Lemma 12, we also have $H \subseteq G^*$. By Lemma 11, for any undirect edge $a—b$ of H_τ where τ is a chain component of H, there exist G_1 and $G_2 \in G^*_{e(a)}$ such that $a \rightarrow b$ occurs in G_1 and $a \leftarrow b$ occurs in G_2. It means that $a—b$ also occurs in $G^*_{e(a)}$. So we have $H \subseteq G^*_{e(a)}$, and then $G^*_{e(a)} = H$. ■

Proof of Theorem 6. By definition of $G^*_{e(V)}$, we have that $G^*_{e(V)}$ has the same skeleton as the essential graph G^* and contains all directed edges of G^*. That is, all directed edges in G^* are also directed in $G^*_{e(V)}$. So property 2 of Theorem 6 holds. Property 3 of Theorem 6 also holds because all DAGs represented by $G^*_{e(V)}$ are Markov equivalent. From Lemmas 12 and 13, we can get that $G^*_{e(V)}$ is a chain graph. ■

Proof of Theorem 7. We first prove property 1. Let $C = ch(V_k) \setminus \tau$. Then $B = ne(V_k) \setminus C$ contains all parents of V_k and the children of V_k in τ. Let $A = An(\{B, V_k\})$ be the ancestor set of all nodes in $\{B, V_k\}$. Since V_i is a parent of V_k for property 1, we have $V_i \in A$. The post-intervention joint distribution of A is

$$P_{V_i}(A) = P'(v_i \mid pa(v_i)) \prod_{v_j \in A \setminus V_i} P(v_j \mid pa(v_j)). \quad (1)$$

Let $U = A \backslash \{B, V_k\}$. Then we have from the post-intervention joint distribution (1)

$$P_{V_i}(v_k|B) = \frac{\sum_U P'(v_i|pa(v_i)) \prod_{v_j \in A \backslash V_i} P(v_j|pa(v_j)))}{\sum_{U,V_k} P'(v_i|pa(v_i)) \prod_{V_j \in A \backslash V_i} P(v_j|pa(v_j))}$$
$$= \frac{\sum_U P'(v_i|pa(v_i)) \prod_{v_j \in A \backslash \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j)) \prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))}{\sum_{U,V_k} P'(v_i|pa(v_i)) \prod_{v_j \in A \backslash \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j)) \prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))},$$

where $\sum_U$ denotes a summation over all variables in the set U.

Below we want to factorize the denominator into a production of summation over U and summation over V_k. First we show that the factor

$$P'(v_i \mid pa(v_i)) \prod_{v_j \in A \backslash \{ch(V_k) \cap \tau, V_k\}} P(v_j \mid pa(v_j))$$

does not contain V_k because V_k appears only in the conditional probabilities of $ch(V_k)$ and the conditional probability of V_k. Next we show that $\prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j \mid pa(v_j))$ does not contain any variable in U. From definition of B, we have $B \supseteq (ch(V_k) \cap \tau)$. Then from definition of U, we have that V_j in $\{ch(V_k) \cap \tau, V_k\}$ is not in U. Now we just need to show that any parent of any node V_j in $\{ch(V_k) \cap \tau, V_k\}$ is also not in U:

1. By definitions of B and U, the parents of V_k is not in U.
2. Consider parents of nodes in $\{ch(V_k) \cap \tau\}$. Let W is such a parent, that is, $W \to V_j$ for $V_j \in \{ch(V_k) \cap \tau\}$. There is a head to head path $(W \to V_j \leftarrow V_k)$. We show that W is not in U separately for two cases: $W \in \tau$ and $W \notin \tau$. For the first case of $W \in \tau$, there is an undirected edge between W and V_k in G^*_τ since there is no v-structure in the subgraph G'_τ for any $G' \in [G]$. Then from definition of B, we have $W \in B$. For the second case of $W \notin \tau$, W must be a parent of V_k by Lemma 10, and then W is in B. Thus we obtain $W \notin U$.

We showed that the factor $\prod_{V_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j \mid pa(v_j))$ does not contain any variable in U. Thus the numerator and the summations over U and V_k in the denominator can be factorized as follows

$$P_{V_i}(v_k|B)$$
$$= \frac{\prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j)) \sum_U P'(v_i|pa(v_i)) \prod_{v_j \in A \backslash \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))}{\sum_{V_k} \prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j)) \sum_U P'(v_i|pa(v_i)) \prod_{v_j \in A \backslash \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))}$$
$$= \frac{\prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))}{\sum_{V_k} \prod_{v_j \in \{ch(V_k) \cap \tau, V_k\}} P(v_j|pa(v_j))} = P(v_k|B).$$

Thus we proved property 1.

Property 2 is obvious since manipulating V_i does not change the distribution of its parent V_k. Formally, let $an(V_k)$ be the ancestor set of V_k. If $V_k \in pa(V_i)$, then we have $P_{V_i}(an(v_k), v_k) = P(an(v_k), v_k)$ and thus $P_{V_i}(V_k) = P(V_k)$. ■

Proof of Theorem 8. Manipulating a node V_i will orient all of undirected edges connecting V_i. Thus the orientations of undirected edges do not depend on the order in

which the variables are manipulated. If a sequence $\mathcal{S}$ is sufficient, then its permutation is also sufficient. ■

Proof of Theorem 9. Suppose that $\mathcal{S} = (V_1, \ldots, V_K)$ is a sufficient set. We delete a node, say V_i, from $\mathcal{S}$, and define $\mathcal{S}'_{[i]} = math\mathcal{S} \setminus \{V_i\}$. If the set $\mathcal{S}'_{[i]}$ is no longer sufficient, then we can add other variables to $\mathcal{S}'_{[i]}$ without adding V_i such that $\mathcal{S}'_{[i]}$ becomes to be sufficient. This is feasible since any undirected edge can be oriented by manipulating either of its two nodes. Thus we have $\bigcap_{i=1}^{K} \mathcal{S}'_{[i]} = \varnothing$. Since all $\mathcal{S}'_{[i]}$ belong to $\mathbb{S}$, we proved $\bigcap_{\mathcal{S}\in\mathbb{S}} \mathcal{S} = \varnothing$.

Similarly, for each minimum sequence $\mathcal{S}$, we can define $\mathcal{S}'_{[i]}$ such that it does not contain V_i and it is a minimum sufficient set. Thus the intersection of all minimum sufficient sets is empty. ■

References

C. Aliferis, I. Tsamardinos, A. Statnikov and L. Brown. Causal explorer: A probabilistic network learning toolkkit for biomedical discovery. In *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 371-376, 2003.

S. A. Andersson, D. Madigan and M. D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505-541, 1997.

R. Castelo and M. D. Perlman. Learning Essential graph Markov models from data. In *Proceedings 1st European Workshop on Probabilistic Graphical Models*, pages 17-24, 2002.

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

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799-805, 2004.

Y. He, Z. Geng and X. Liang. Learning causal structures based on Markov equivalence class. In *ALT, Lecture Notes in Artificial Intelligence* 3734, pages 92-106, 2005.

D. Heckerman, D. Geiger and D. M. Chickering. Learning Bayesian networks: The Combination of knowledge and statistical data. *Machine Learning*, 20:197-243, 1995.

D. Heckerman. A Bayesian approach to causal discovery. *Data Mining and Knowledge Discovery*, 1(1):79-119, 1997.

R. Jansen, H. Y. Yu and D. Greenbaum. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449-453, 2003.

S. L. Lauritzen. *Graphical Models*. Oxford Univ. Press. 1996.

S. L. Lauritzen, T. S. Richardson. Chain graph models and their casual interpretations. *Journal of the Royal Statistical society series B-statistical methodology*,64:321-348, Part 3, 2002.

M. Kalisch, P. Buhlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm.*Journal of Machine Learning Research* 8, 613-636, 2007.

K. P. Murphy. Active Learning of Causal Bayes Net Structure, *Technical Report*, Department of Computer Science, University of California Berkeley, 2001.

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

J. Pearl. Graphical models, causality and intervention. *Statist. Sci.*, 8:266-269, 1993.

J. Pearl. Causal inference from indirect experiments. *Artifcal Intelligence in Medicine*, 7:561-582, 1995.

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

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

J. Tian and J. Pearl. Causal Discovery from Changes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 512-521, 2001a.

J. Tian and J. Pearl. Causal Discovery from Changes: a Bayesian Approach, UCLA Cognitive Systems Laboratory, Technical Report (R-285), 2001b.

S. Tong and D. Koller. Active learning for structure in bayesian networks. In *International Joint Conference on Artificial Intelligence*, pages 863-869, 2001.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 220-227, 1990.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, New York. 1990.

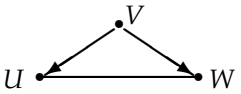


Figure 5: SG_1

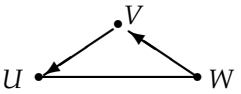


Figure 6: SG_2

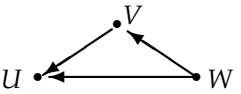


Figure 7: SG_3

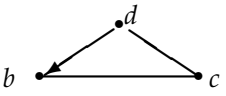


Figure 8: SG_6

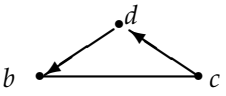


Figure 9: SG_{6_1}

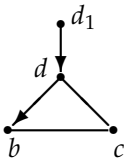


Figure 10: SG_7

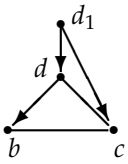


Figure 11: SG_8

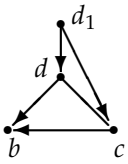


Figure 12: SG_9

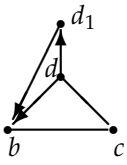


Figure 13: SG_{10}

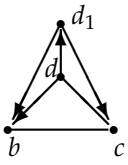


Figure 14: SG_{11}

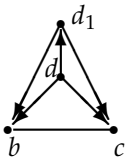


Figure 15: SG_{12}

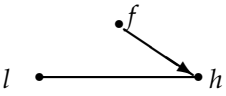


Figure 16: SG_4

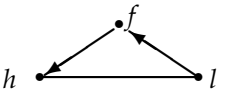


Figure 17: SG_5

JMLR: Workshop and Conference Proceedings 10:41–70, 2009 Submitted 7/07; Revised 9/08; Published 1/09

Markov Properties for Linear Causal Models with Correlated Errors

Changsung Kang CSKANG@CS.IASTATE.EDU
Jin Tian JTIAN@CS.IASTATE.EDU
Department of Computer Science
Iowa State University
Ames, IA 50011, USA

Editor: André Elisseeff

Abstract

A linear causal model with correlated errors, represented by a DAG with bi-directed edges, can be tested by the set of conditional independence relations implied by the model. A global Markov property specifies, by the d-separation criterion, the set of all conditional independence relations holding in any model associated with a graph. A local Markov property specifies a much smaller set of conditional independence relations which will imply all other conditional independence relations which hold under the global Markov property. For DAGs with bi-directed edges associated with arbitrary probability distributions, a local Markov property is given in Richardson (2003) which may invoke an exponential number of conditional independencies. In this paper, we show that for a class of linear structural equation models with correlated errors, there is a local Markov property which will invoke only a linear number of conditional independence relations. For general linear models, we provide a local Markov property that often invokes far fewer conditional independencies than that in Richardson (2003). The results have applications in testing linear structural equation models with correlated errors.

Keywords: Markov properties, linear causal models, linear structural equation models, graphical models

1. Introduction

Linear causal models called structural equation models (SEMs) are widely used for causal reasoning in social sciences, economics, and artificial intelligence (Goldberger, 1972; Bollen, 1989; Spirtes et al., 2001; Pearl, 2000). One important problem in the applications of linear causal models is testing a hypothesized model against the given data. While the conventional method involves maximum likelihood estimation of the covariance matrix, an alternative approach has been proposed recently which involves testing for the conditional independence relationships implied by the model (Spirtes et al., 1998; Pearl, 1998; Pearl and Meshkat, 1999; Pearl, 2000; Shipley, 2000, 2003). The advantages of using this new test method instead of the traditional global fitting test have been discussed in Pearl (1998), Shipley (2000), McDonald (2002) and Shipley (2003). The method can be applied in small data samples and it can test "local" features of the model.

To apply this test method, one needs to be able to identify the conditional independence relationships implied by an SEM. This can be achieved by representing the

SEM with a graph called a path diagram (Wright, 1934) and then reading independence relations from the path diagram. For a linear SEM without correlated errors, the corresponding path diagram is a directed acyclic graph (DAG). The set of all conditional independence relations holding in any model associated with a DAG, often called a global Markov property for the DAG, can be read by the d-separation criterion (Pearl, 1988). However, it is not necessary to test for all the independencies implied by the model as a subset of those independencies may imply all others. A local Markov property specifies a much smaller set of conditional independence relations which will imply (using the laws of probability) all other conditional independence relations that hold under the global Markov property. A well-known local Markov property for DAGs is that each variable is conditionally independent of its non-descendants given its parents (Lauritzen et al., 1990; Lauritzen, 1996). Based on this local Markov property, Pearl and Meshkat (1999) and Shipley (2000) proposed testing methods for linear SEMs without correlated errors that involve at most one conditional independence test for each pair of variables.

On the other hand, the path diagrams for linear SEMs with correlated errors are DAGs with bi-directed edges ($\leftrightarrow$) where bi-directed edges are used to represent correlated errors. A DAG with bi-directed edges is called an *acyclic directed mixed graph (ADMG)* in Richardson (2003). The set of all conditional independence relations encoded in an ADMG can still be read by (a natural extension of) the d-separation criterion (called m-separation in Richardson, 2003) which provides the global Markov property for ADMGs (Spirtes et al., 1998; Koster, 1999; Richardson, 2003). A local Markov property for ADMGs is given in Richardson (2003), which, in the worst case, may invoke an exponential number of conditional independence relations, a sharp difference with the local Markov property for DAGs, where only one conditional independence relation is associated with each variable. Shipley (2003) suggested a method for testing linear SEMs with correlated errors but the method may or may not, depending on the actual models, be able to find a subset of conditional independence relations that imply all others.

In this paper, we seek to improve the local Markov property given in Richardson (2003) for linear SEMs with correlated errors. The local Markov property in Richardson (2003) is applicable for ADMGs associated with arbitrary probability distributions. Specifically, only semi-graphoid axioms which must hold in all probability distributions (Pearl, 1988) are used in showing that the set of conditional independence relations specified by the local Markov property will imply all those specified by the global Markov property. On the other hand, in linear SEMs, variables are assumed to have normal distributions, and it is known that normal distributions also satisfy the so-called composition axiom. Therefore, in this paper, we look for local Markov properties for ADMGs associated with probability distributions that satisfy the composition axiom. We will show that for a class of ADMGs, the local Markov property will invoke only one conditional independence relation for each variable, and therefore testing for the corresponding linear SEMs will involve at most one conditional independence test for each pair of variables. For general ADMGs, we provide a procedure that reduces the number of conditional independencies invoked by the local Markov property given in Richardson (2003), and therefore reduces the complexity of testing linear SEMs with correlated errors.

In the test of conditional independence relations, the efficiency of the test is influenced by the size of the conditioning set (that is, the number of conditioning variables) with a small conditioning set having advantage over a large one. The conditional

independence relations invoked by the standard local Markov property for DAGs use a parent set as the conditioning set. Pearl and Meshkat (1999) have shown for linear SEMs without correlated errors how to find a set of conditional independence relations that may involve fewer conditioning variables. In this paper, we also generalize this result to linear SEMs with correlated errors.

The paper is organized as follows. In Section 2, we introduce linear SEMs, give basic notation and definitions, and present the local Markov property developed in Richardson (2003). In Section 3, we show that for a class of ADMGs, there is a local Markov property for probability distributions satisfying the composition axiom that invokes only a linear number of conditional independence relations. We also show a local Markov property that may involve fewer conditioning variables. In Section 4, we consider general ADMGs (for probability distributions satisfying the composition axiom) and show a local Markov property that invokes fewer conditional independencies than that in Richardson (2003). Section 5 concludes the paper.

2. Preliminaries and Motivation

In this section, we give basic definitions and introduce some relevant concepts.

2.1. Linear Causal Models

The SEM technique was developed by geneticists (Wright, 1934) and economists (Haavelmo, 1943) for assessing cause-effect relationships from a combination of statistical data and qualitative causal assumptions. It is an important causal analysis tool widely used in social sciences, economics, and artificial intelligence (Goldberger, 1972; Duncan, 1975; Bollen, 1989; Spirtes et al., 2001). For a review of SEMs and causality we refer to Pearl (1998).

In an SEM, the causal relationships among a set of variables are often assumed to be linear and expressed by linear equations. Each equation describes the dependence of one variable in terms of the others. For example, an equation

$$Y = aX + \epsilon \tag{1}$$

represents that X may have a *direct* causal influence on Y and that no other variables have (direct) causal influences on Y except those factors (represented by the error term ϵ traditionally assumed to have normal distribution) that are omitted from the model. The parameter a quantifies the (direct) causal effect of X on Y. An equation like (1) with a causal interpretation represents an autonomous causal mechanism and is said to be *structural*.

As an example, consider the following model from Pearl (2000) that concerns the relations between smoking (X) and lung cancer (Y), mediated by the amount of tar (Z) deposited in a person's lungs:

$$\begin{aligned} X &= \epsilon_1, \\ Z &= aX + \epsilon_2, \\ Y &= bZ + \epsilon_3. \end{aligned}$$

The model assumes that the amount of tar deposited in the lungs depends on the level of smoking (and external factors) and that the production of lung cancer depends on the amount of tar in the lungs but smoking has no effect on lung cancer except as mediated

through tar deposits. To fully specify the model, we also need to decide whether those omitted factors (ϵ_1, ϵ_2, ϵ_3) are correlated or not. We may assume that no other factor that affects tar deposit is correlated with the omitted factors that affect smoking or lung cancer ($Cov(\epsilon_1, \epsilon_2) = Cov(\epsilon_2, \epsilon_3) = 0$). However, there might be unobserved factors (say some unknown carcinogenic genotype) that affect both smoking and lung cancer ($Cov(\epsilon_1, \epsilon_3) \neq 0$), but the genotype nevertheless has no effect on the amount of tar in the lungs except indirectly (through smoking). Often, it is illustrative to express our qualitative causal assumptions in terms of a graphical representation, as shown in Figure 1.

We now formally define the model that we will consider in this paper. A *linear causal model* (or *linear SEM*) over a set of random variables $V = \{V_1, \ldots, V_n\}$ is given by a set of structural equations of the form

$$V_j = \sum_i c_{ji} V_i + \epsilon_j, \; j = 1, \ldots, n, \tag{2}$$

where the summation is over the variables in V judged to be immediate causes of V_j. c_{ji}, called a *path coefficient*, quantifies the direct causal influence of V_i on V_j. ϵ_j's represent "error" terms due to omitted factors and are assumed to have normal distribution. We consider recursive models and assume that the summation in (2) is for $i < j$, that is, $c_{ji} = 0$ for $i \geq j$.

We denote the covariances between observed variables $\sigma_{ij} = Cov(V_i, V_j)$, and between error terms $\psi_{ij} = Cov(\epsilon_i, \epsilon_j)$. We denote the following matrices, $\Sigma = [\sigma_{ij}]$, $\Psi = [\psi_{ij}]$, and $C = [c_{ij}]$. The parameters of the model are the non-zero entries in the matrices C and Ψ. A parameterization of the model assigns a value to each parameter in the model, which then determines a unique covariance matrix Σ given by (see, for example, Bollen, 1989)

$$\Sigma = (I - C)^{-1} \Psi ((I - C)^t)^{-1}.$$

The structural assumptions encoded in the model are the zero path coefficients and zero error covariances. The model structure can be represented by a DAG G with (dashed) bi-directed edges (an ADMG), called a *causal diagram* (or *path diagram*), as follows: the nodes of G are the variables $V_1, \ldots, V_n$; there is a directed edge from V_i to V_j in G if V_i appears in the structural equation for V_j, that is, $c_{ji} \neq 0$; there is a bi-directed edge between V_i and V_j if the error terms ϵ_i and ϵ_j have non-zero correlation. For example, the smoking-and-lung-cancer SEM is represented by the causal diagram in Figure 1, in which each directed edge is annotated by the corresponding path coefficient.

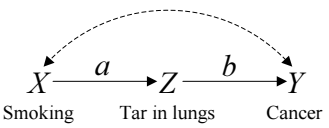


Figure 1: Causal diagram illustrating the effect of smoking on lung cancer

We note that linear SEMs are often used without explicit causal interpretation. A linear SEM in which error terms are uncorrelated consists of a set of regression equations. Note that an equation as given by (2) is a regression equation if and only if ϵ_j is uncorrelated with each V_i ($Cov(V_i, \epsilon_j) = 0$). Hence, an equation in an SEM

with correlated errors may not be a regression equation. Linear SEMs provide a more powerful way to model data than the regression models taking into account correlated error terms.

2.2. Model Testing and Markov Properties

One important task in the applications of linear SEMs is to test a model against data. One approach for this task is to test for the conditional independence relationships implied by the model, which can be read from the causal diagram by the d-separation criterion as defined in the following.[1] A *path* between two vertices V_i and V_j in an ADMG consists of a sequence of consecutive edges of any type (directed or bi-directed). A vertex V_i is said to be an *ancestor* of a vertex V_j if there is a path $V_i \to \cdots \to V_j$. A non-endpoint vertex W on a path is called a *collider* if two arrowheads on the path meet at W, that is, $\to W \leftarrow$, $\leftrightarrow W \leftrightarrow$, $\leftrightarrow W \leftarrow$, $\to W \leftrightarrow$; all other non-endpoint vertices on a path are *non-colliders*, that is, $\leftarrow W \to$, $\leftarrow W \leftarrow$, $\to W \to$, $\leftrightarrow W \to$, $\leftarrow W \leftrightarrow$. A path between vertices V_i and V_j in an ADMG is said to be *d-connecting given a set* of vertices Z if

1. every non-collider on the path is not in Z, and
2. every collider on the path is an ancestor of a vertex in Z.

If there is no path d-connecting V_i and V_j given Z, then V_i and V_j are said to be *d-separated* given Z. Sets X and Y are said to be *d-separated* given Z, if for every pair V_i, V_j, with $V_i \in X$ and $V_j \in Y$, V_i and V_j are d-separated given Z. Let $I(X, Z, Y)$ denote that X is conditionally independent of Y given Z. The set of all the conditional independence relations encoded by a causal diagram G is specified by the following global Markov property.

Definition 1 (The Global Markov Property (GMP)) *A probability distribution P is said to satisfy the global Markov property for G if for arbitrary disjoint sets X, Y, Z with X and Y being nonempty,*

$$\text{(GMP)} \qquad X \text{ is d-separated from } Y \text{ given } Z \text{ in } G \Longrightarrow I(X, Z, Y).$$

The global Markov property typically involves a vast number of conditional independence relations and it is possible to test for a subset of those independencies that will imply all others. A local Markov property specifies a much smaller set of conditional independence relations which will imply by the laws of probability all other conditional independence relations that hold under the global Markov property. For example, a well-known local Markov property for DAGs is that each variable is conditionally independent of its non-descendants given its parents. The causal diagram for a linear SEM with correlated errors is an ADMG and a local Markov property for ADMGs is given in Richardson (2003).

Note that in linear SEMs, the conditional independence relations will correspond to zero partial correlations (Lauritzen, 1996):

$$\rho_{V_i V_j . Z} = 0 \Longleftrightarrow I(\{V_i\}, Z, \{V_j\}).$$

1. The d-separation criterion was originally defined for DAGs (Pearl, 1988) but can be naturally extended for ADMGs and is called m-separation in Richardson (2003).

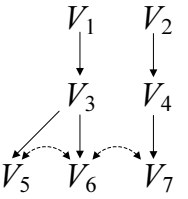


Figure 2: A causal diagram

As an example, for the linear SEM with the causal diagram in Figure 2, if we use the local Markov property in Richardson (2003), then we need to test for the vanishing of the following set of partial correlations (for ease of notation, we write $\rho_{ij.Z}$ to denote $\rho_{V_iV_j.Z}$):

$$\{\rho_{21}, \rho_{32.1}, \rho_{43.2}, \rho_{41.2}, \rho_{54.3}, \rho_{52.3}, \rho_{51.3}, \rho_{64.53}, \rho_{62.53}, \rho_{61.53}, \rho_{64.3}, \rho_{62.3}, \rho_{61.3}, \rho_{72.6543}, \rho_{71.6543}, \rho_{72.643}, \rho_{71.643}, \rho_{75.4}, \rho_{73.4}, \rho_{72.4}, \rho_{71.4}\}. \tag{3}$$

The local Markov property in Richardson (2003) is valid for any probability distributions. In fact, the equivalence of the global and local Markov properties is proved using the following so-called *semi-graphoid axioms* (Pearl, 1988) that probabilistic conditional independencies must satisfy:

- Symmetry
$$I(X, Z, Y) \iff I(Y, Z, X).$$
- Decomposition
$$I(X, Z, Y \cup W) \implies I(X, Z, Y) \ \& \ I(X, Z, W).$$
- Weak Union
$$I(X, Z, Y \cup W) \implies I(X, Z \cup W, Y).$$
- Contraction
$$I(X, Z, Y) \ \& \ I(X, Z \cup Y, W) \implies I(X, Z, Y \cup W).$$

where X, Y, Z, and W are disjoint sets of variables.

On the other hand, in linear SEMs the variables are assumed to have normal distributions, and normal distributions also satisfy the following *composition* axiom:

- Composition
$$I(X, Z, Y) \ \& \ I(X, Z, W) \implies I(X, Z, Y \cup W).$$

Therefore, we expect a local Markov property for linear SEMs to invoke fewer conditional independence relations than that for arbitrary distributions. In this paper, we will derive reduced local Markov properties for linear SEMs by making use of the composition axiom. As an example, for the linear SEM in Figure 2, a local Markov property which we will present in this paper (see Section 3.3) says that we only need to test for the vanishing of the following set of partial correlations:

$$\{\rho_{21}, \rho_{32}, \rho_{43}, \rho_{41}, \rho_{54}, \rho_{52}, \rho_{51.3}, \rho_{64}, \rho_{62}, \rho_{61.3}, \rho_{75}, \rho_{73}, \rho_{71}, \rho_{72.4}\}.$$

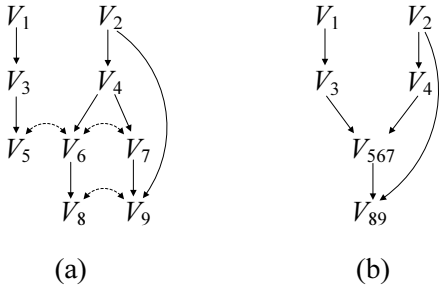


Figure 3: An ADMG and its compressed graph

The number of tests needed and the size of the conditioning set Z are both substantially reduced compared with (3), thus leading to a more economical way of testing the given model.

2.3. A Local Markov Property for ADMGs

In this section, we describe the local Markov property for ADMGs associated with arbitrary probability distributions presented in Richardson (2003). In this paper, this Markov property will be used as an important tool to prove the equivalence of our local Markov properties and the global Markov property.

First, we define some graphical notations. For a vertex X in an ADMG G, $\mathrm{pa}_G(X) \equiv \{Y | Y \to X \text{ in } G\}$ is the set of *parents* of X. $\mathrm{sp}_G(X) \equiv \{Y | Y \leftrightarrow X \text{ in } G\}$ is the set of *spouses* of X. $\mathrm{an}_G(X) \equiv \{Y | Y \to \cdots \to X \text{ in } G \text{ or } Y = X\}$ is the set of *ancestors* of X. And $\mathrm{de}_G(X) \equiv \{Y | Y \leftarrow \cdots \leftarrow X \text{ in } G \text{ or } Y = X\}$ is the set of *descendants* of X. These definitions will be applied to sets of vertices, so that, for example, $\mathrm{pa}_G(A) \equiv \cup_{X \in A} \mathrm{pa}_G(X)$, $\mathrm{sp}_G(A) \equiv \cup_{X \in A} \mathrm{sp}_G(X)$, etc.

Definition 2 (C-component) *A c-component of G is a maximal set of vertices in G such that any two vertices in the set are connected by a path on which every edge is of the form $\leftrightarrow$; a vertex that is not connected to any bi-directed edge forms a c-component by itself.*

For example, the ADMG in Figure 3 (a) is composed of 6 c-components $\{V_1\}$, $\{V_2\}$, $\{V_3\}$, $\{V_4\}$, $\{V_5, V_6, V_7\}$ and $\{V_8, V_9\}$. The *district* of X in G is the c-component of G that includes X. Thus,

$$\mathrm{dis}_G(X) \equiv \{Y | Y \leftrightarrow \cdots \leftrightarrow X \text{ in } G \text{ or } Y = X\}.$$

For example, in Figure 3 (a), we have $\mathrm{dis}_G(V_5) = \{V_5, V_6, V_7\}$ and $\mathrm{dis}_G(V_8) = \{V_8, V_9\}$. A set A is said to be *ancestral* if it is closed under the ancestor relation, that is, if $\mathrm{an}_G(A) = A$. Let G_A denote the induced subgraph of G on the vertex set A, formed by removing from G all vertices that are not in A, and all edges that do not have both endpoints in A.

Definition 3 (Markov Blanket)[2] *If A is an ancestral set in an ADMG G, and X is a vertex in A that has no children in A then the* Markov blanket of vertex X with respect to the

2. The definition of Markov blanket here follows that in Richardson (2003) and is compatible with that in Pearl (1988).

induced subgraph on A, *denoted* $\mathrm{mb}(X, A)$ *is defined to be*

$$\mathrm{mb}(X, A) \equiv \mathrm{pa}_{G_A}\left(\mathrm{dis}_{G_A}(X)\right) \cup \left(\mathrm{dis}_{G_A}(X) \setminus \{X\}\right).$$

For example, for an ancestral set $A = \mathrm{an}_G(\{V_5, V_6\}) = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ in Figure 3 (a), we have

$$\mathrm{mb}(V_5, A) = \{V_3, V_4, V_6\}.$$

An ordering $(\prec)$ on the vertices of G is said to be consistent with G if $X \prec Y \Rightarrow Y \notin \mathrm{an}_G(X)$. Given a consistent ordering $\prec$, let $\mathrm{pre}_{G,\prec}(X) \equiv \{Y | Y \prec X \text{ or } Y = X\}$.

Definition 4 (The Ordered Local Markov Property (LMP,$\prec$)) *A probability distribution P satisfies the ordered local Markov property for G with respect to a consistent ordering $\prec$, if, for any X and ancestral set A such that $X \in A \subseteq \mathrm{pre}_{G,\prec}(X)$,*

$$\text{(LMP,}\prec\text{)} \qquad I(\{X\}, \mathrm{mb}(X, A), A \setminus (\mathrm{mb}(X, A) \cup \{X\})). \tag{4}$$

Theorem 5 (Richardson, 2003) *If G is an ADMG and $\prec$ is a consistent ordering, then a probability distribution P satisfies the ordered local Markov property for G with respect to $\prec$ if and only if P satisfies the global Markov property for G.*

We will write (GMP) $\Longleftrightarrow$ (LMP,$\prec$) to denote the equivalence of the two Markov properties. Therefore the (smaller) set of conditional independencies specified in the ordered local Markov property will imply all other conditional independencies which hold under the global Markov property. It is possible to further reduce the number of conditional independence relations in the ordered local Markov property. An ancestral set A, with $X \in A \subseteq \mathrm{pre}_{G,\prec}(X)$ is said to be *maximal with respect to the Markov blanket* $\mathrm{mb}(X, A)$ if, whenever there is a set B such that $A \subseteq B \subseteq \mathrm{pre}_{G,\prec}(X)$ and $\mathrm{mb}(X, A) = \mathrm{mb}(X, B)$, then $A = B$. For example, suppose that we are given an ordering $\prec$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$ for the graph G in Figure 3 (a). While an ancestral set $A = \mathrm{an}_G(\{V_3, V_6, V_7\}) = \{V_1, V_2, V_3, V_4, V_6, V_7\}$ is maximal with respect to the Markov blanket $\mathrm{mb}(V_7, A) = \{V_4, V_6\}$, an ancestral set $A' = \mathrm{an}_G(\{V_6, V_7\}) = \{V_2, V_4, V_6, V_7\}$ is not. It was shown that we only need to consider ancestral sets A which are maximal with respect to $\mathrm{mb}(X, A)$ in the ordered local Markov property (Richardson, 2003). Thus, we will consider only maximal ancestral sets A when we discuss (LMP,$\prec$) for the rest of this paper. The following lemma characterizes maximal ancestral sets.

Lemma 6 (Richardson, 2003) *Let X be a vertex and A an ancestral set in G with consistent ordering $\prec$ such that $X \in A \subseteq \mathrm{pre}_{G,\prec}(X)$. The set A is maximal with respect to the Markov blanket* mb*(X,A) if and only if*

$$A = \mathrm{pre}_{G,\prec}(X) \setminus \mathrm{de}_G(\mathrm{h}(X, A))$$

where

$$\mathrm{h}(X, A) \equiv \mathrm{sp}_G\Big(\mathrm{dis}_{G_A}(X)\Big) \setminus \Big(\{X\} \cup \mathrm{mb}(X, A)\Big).$$

Even though we only consider maximal ancestral sets, the ordered local Markov property may still invoke an exponential number of conditional independence relations. For example, for a vertex X, if $\mathrm{dis}_G(X) \subseteq \mathrm{pre}_{G,\prec}(X)$ and $\mathrm{dis}_G(X)$ has a clique of n vertices joined by bi-directed edges, then there are at least $O(2^{n-1})$ different Markov blankets.

It should be noted that only the semi-graphoid axioms were used to prove Theorem 5 on the equivalence of the two Markov properties and no assumptions about probability distributions were made. Next we will show that the ordered local Markov property can be further reduced if we use the composition axiom in addition to the semi-graphoid axioms. The local Markov properties we obtained (in Sections 3 and 4) are not restricted to linear causal models in that they are actually valid for any probability distributions that satisfy the composition axiom.

3. Markov Properties for ADMGs without Directed Mixed Cycles

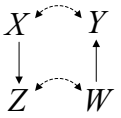


Figure 4: Directed mixed cycles

In this section, we introduce three local Markov properties for a class of ADMGs and show that they are equivalent to the global Markov property. Also, we discuss related work in maximal ancestral graphs and chain graphs. First, we give some definitions.

Definition 7 (Directed Mixed Cycle) *A path is said to be a directed mixed path from X to Y if it contains at least one directed edge and every edge on the path is either of the form $Z \leftrightarrow W$, or $Z \rightarrow W$ with W between Z and Y. A directed mixed path from X to Y together with an edge $Y \rightarrow X$ or $Y \leftrightarrow X$ is called a directed mixed cycle.*

For example, the path $X \rightarrow Z \leftrightarrow W \rightarrow Y \leftrightarrow X$ in the graph in Figure 4 forms a directed mixed cycle. In this section, we will consider only ADMGs without directed mixed cycles.

Definition 8 (Compressed Graph) *Let G be an ADMG. The compressed graph of G is defined to be the graph $G' = (V', E')$, $V' = \{V_C \mid C \text{ is a c-component of } G\}$, $E' = \{V_{C_i} \rightarrow V_{C_j} \mid \text{there is an edge } X \rightarrow Y \text{ in } G \text{ such that } X \in C_i, Y \in C_j\}$.*

Figure 3 shows an ADMG and its compressed graph. If there exists a directed mixed cycle in an ADMG G, there will be a cycle or a self-loop in the compressed graph of G. For example, if for two vertices X and Y in a c-component C of G there exists an edge $X \rightarrow Y$, then the compressed graph of G contains a self-loop $\overset{\curvearrowright}{V_C}$. The following proposition holds.

Proposition 9 *Let G be an ADMG. The compressed graph of G is a DAG if and only if G has no directed mixed cycles.*

3.1. The Reduced Local Markov Property

In this section, we introduce a local Markov property for ADMGs without directed mixed cycles which only invokes a linear number of conditional independence relations and show that it is equivalent to the global local Markov property.

Definition 10 (The Reduced Local Markov Property (RLMP)) *Let G be an ADMG without directed mixed cycles. A probability distribution P is said to satisfy the reduced local Markov property for G if*

$$\text{(RLMP)} \qquad \forall X \in V, \quad I(\{X\}, \text{pa}_G(X), V \setminus \text{f}(X, G)) \tag{5}$$

where $\text{f}(X, G) \equiv \text{pa}_G(X) \cup \text{de}_G(\{X\} \cup \text{sp}_G(X))$.

The reduced local Markov property states that *a variable is independent of the variables that are neither its descendants nor its spouses' descendants given its parents*.

Theorem 11 *If a probability distribution P satisfies the composition axiom and an ADMG G has no directed mixed cycles, then*

$$\text{(GMP)} \iff \text{(RLMP)}.$$

Proof: $\text{(GMP)} \implies \text{(RLMP)}$
We need to prove that any variable X is d-separated from $V \setminus \text{f}(X, G)$ given $\text{pa}_G(X)$ in G with no directed mixed cycle. Consider a vertex $\alpha \in V \setminus \text{f}(X, G)$. We will show that there is no path d-connecting X and α given $\text{pa}_G(X)$. There are four possible cases for any path between X and α.

1. $X \leftarrow \beta \cdots \alpha$
2. $X \rightarrow \cdots \rightarrow \delta \leftarrow * \cdots \alpha$
3. $X \leftrightarrow \gamma \leftarrow * \cdots \alpha$
4. $X \leftrightarrow \gamma \rightarrow \cdots \rightarrow \delta \leftarrow * \cdots \alpha$

A symbol $*$ serves as a wildcard for an end of an edge. For example, $\leftarrow *$ represents both $\leftarrow$ and $\leftrightarrow$. In case 1, $\beta \in \text{pa}_G(X)$. In case 2, the collider δ is not an ancestor of a vertex in $\text{pa}_G(X)$ (otherwise, there would be a cycle). In cases 3 and 4, neither γ nor δ is an ancestor of a vertex in $\text{pa}_G(X)$ (otherwise, there would be directed mixed cycles). In any case, the path is not d-connecting given $\text{pa}_G(X)$. ■

Proof: $\text{(RLMP)} \implies \text{(GMP)}$
We will show that for some consistent ordering $\prec$, $\text{(RLMP)} \implies \text{(LMP}, \prec)$. Then, by Theorem 5, we have $\text{(RLMP)} \implies \text{(GMP)}$.

We construct a consistent ordering with the desired property as follows.

1. Construct the compressed graph G' of G.
2. Let $\prec'$ be any consistent ordering on G'. Construct a consistent ordering $\prec$ from $\prec'$ by replacing each V_C (corresponding to each c-component C of G) in $\prec'$ with the vertices in C (the ordering of the vertices in C is arbitrary).

We now prove that (RLMP) $\implies$ (LMP,$\prec$). Assume that a probability distribution P satisfies (RLMP). Consider the set of conditional independence relations invoked by (LMP,$\prec$) for each variable X given in (4). First, observe that for any vertex Y in $\text{dis}_{G_A}(X)$, we have

$$A \setminus (\text{pa}_G(Y) \cup \{Y\} \cup \text{sp}_G(Y)) \subseteq V \setminus \text{f}(Y, G),$$

since

$$\begin{aligned} & A \setminus (\mathrm{pa}_G(Y) \cup \{Y\} \cup \mathrm{sp}_G(Y)) \\ & = A \setminus \Big(\big(\mathrm{pa}_G(Y) \cup \{Y\} \cup \mathrm{sp}_G(Y) \big) \cup \big(\mathrm{de}_G(\{Y\} \cup \mathrm{sp}_G(Y)) \setminus (\{Y\} \cup \mathrm{sp}_G(Y)) \big) \Big) \quad (6) \\ & = A \setminus \mathrm{f}(Y, G). \end{aligned}$$

The equality (6) holds since the vertices in $\mathrm{de}_G(\{Y\} \cup \mathrm{sp}_G(Y)) \setminus (\{Y\} \cup \mathrm{sp}_G(Y))$ do not appear in A (because of the way $\prec$ is constructed, no descendant of $\mathrm{dis}_{G_A}(X)$ is in A). Thus, by (5), for all Y in $\mathrm{dis}_{G_A}(X)$, we have

$$I(\{Y\}, \mathrm{pa}_G(Y), A \setminus (\mathrm{pa}_G(Y) \cup \{Y\} \cup \mathrm{sp}_{G_A}(Y))).$$

Let $S_1 = \mathrm{pa}_G(\mathrm{dis}_{G_A}(X)) \setminus \mathrm{pa}_G(Y)$ and $S_2 = A \setminus (\mathrm{mb}(X, A) \cup \{X\})$. It follows that

$$\begin{aligned} & S_1 \subseteq A \setminus (\mathrm{pa}_G(Y) \cup \{Y\} \cup \mathrm{sp}_G(Y)) \text{ and} \\ & S_2 \subseteq A \setminus (\mathrm{pa}_G(Y) \cup \{Y\} \cup \mathrm{sp}_G(Y)). \end{aligned}$$

Also, we have

$$S_1 \cap S_2 = \emptyset,$$

since $S_1 \subseteq \mathrm{mb}(X, A)$. Therefore, for $Y \in \mathrm{dis}_{G_A}(X)$,

$I(\{Y\}, \mathrm{pa}_G(Y), S_1 \cup S_2)$	by decomposition
$I(\{Y\}, \mathrm{pa}_G(Y) \cup S_1, S_2)$	by weak union
$I(\mathrm{dis}_{G_A}(X), \mathrm{pa}_G(\mathrm{dis}_{G_A}(X)), A \setminus (\mathrm{mb}(X, A) \cup \{X\}))$	by composition
$I(\{X\}, \mathrm{pa}_G(\mathrm{dis}_{G_A}(X)) \cup (\mathrm{dis}_{G_A}(X) \setminus \{X\}),$ $A \setminus (\mathrm{mb}(X, A) \cup \{X\}))$	by weak union.

Thus, we have

$$I(\{X\}, \mathrm{mb}(X, A), A \setminus (\mathrm{mb}(X, A) \cup \{X\}))$$

by the definition of the Markov blanket of X with respect to A. ■

As an example, consider the ADMG G in Figure 3 (a) which has no directed mixed cycles. The graph in Figure 3 (b) is the compressed graph G' of G described in the proof. From the ordering $\prec'$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_{567} \prec V_{89}$, we obtain the ordering $\prec$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$. The ordered local Markov property (LMP,$\prec$) involves the following conditional independence relations:

$$\begin{aligned} & I(\{V_2\}, \emptyset, \{V_1\}), && I(\{V_3\}, \{V_1\}, \{V_2\}), \\ & I(\{V_4\}, \{V_2\}, \{V_1, V_3\}), && I(\{V_5\}, \{V_3\}, \{V_1, V_2, V_4\}), \\ & I(\{V_6\}, \{V_3, V_4, V_5\}, \{V_1, V_2\}), && I(\{V_6\}, \{V_4\}, \{V_1, V_2, V_3\}), \\ & I(\{V_7\}, \{V_3, V_4, V_5, V_6\}, \{V_1, V_2\}), && I(\{V_7\}, \{V_4, V_6\}, \{V_1, V_2, V_3\}), \\ & I(\{V_7\}, \{V_4\}, \{V_1, V_2, V_3, V_5\}), && I(\{V_8\}, \{V_6\}, \{V_1, V_2, V_3, V_4, V_5, V_7\}), \\ & I(\{V_9\}, \{V_2, V_6, V_7, V_8\}, \{V_1, V_3, V_4, V_5\}), && I(\{V_9\}, \{V_2, V_7\}, \{V_1, V_3, V_4, V_5, V_6\}). \quad (7) \end{aligned}$$

(RLMP) invokes the following conditional independence relations:

$$
\begin{aligned}
&I(\{V_1\},\emptyset,\{V_2,V_4,V_6,V_7,V_8,V_9\}), && I(\{V_2\},\emptyset,\{V_1,V_3,V_5\}),\\
&I(\{V_3\},\{V_1\},\{V_2,V_4,V_6,V_7,V_8,V_9\}), && I(\{V_4\},\{V_2\},\{V_1,V_3,V_5\}),\\
&I(\{V_5\},\{V_3\},\{V_1,V_2,V_4,V_7,V_9\}), && I(\{V_6\},\{V_4\},\{V_1,V_2,V_3\}),\\
&I(\{V_7\},\{V_4\},\{V_1,V_2,V_3,V_5\}), && I(\{V_8\},\{V_6\},\{V_1,V_2,V_3,V_4,V_5,V_7\}),\\
&I(\{V_9\},\{V_2,V_7\},\{V_1,V_3,V_4,V_5,V_6\}) && \qquad (8)
\end{aligned}
$$

which, by Theorem 11, imply all the conditional independence relations in (7).

For the special case of graphs containing only bi-directed edges,[3] Kauermann (1996) provides a local Markov property for probability distributions obeying the composition axiom as follows:

$$\forall X \in V, \quad I(\{X\},\emptyset,V\setminus(\{X\}\cup \mathrm{sp}_G(X))). \tag{9}$$

Since a graph containing only bi-directed edges is a special case of ADMGs without directed mixed cycles, the reduced local Markov property (RLMP) is applicable, and it turns out that (RLMP) reduces to (9) for graphs containing only bi-directed edges. Therefore (RLMP) includes the local Markov property given in Kauermann (1996) as a special case.

3.2. The Ordered Reduced Local Markov Property

The set of zero partial correlations corresponding to a conditional independence relation $I(X,Z,Y)$ is

$$\{\rho_{V_iV_j\cdot Z}=0 \mid V_i\in X, V_j\in Y\}.$$

Although (RLMP) gives only a linear number of conditional independence relations, the number of zero partial correlations may be larger than that invoked by (LMP,$\prec$) in some cases. For example, 12 conditional independence relations in (7) involve 37 zero partial correlations while 9 conditional independence relations in (8) involve 41 zero partial correlations. In this section, we will show an ordered local Markov property such that at most one zero partial correlation is invoked for each pair of variables.

Definition 12 (C-ordering) *Let G be an ADMG. A consistent ordering $\prec$ on the vertices of G is said to be a c-ordering if all the vertices in each c-component of G are consecutively ordered in $\prec$.*

For example, the ordering $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$ is a c-ordering on the vertices of G in Figure 3 (a). The following holds.

Proposition 13 *There exists a c-ordering on the vertices of G if G does not have directed mixed cycles.*

We can easily construct a c-ordering from the compressed graph of G. We introduce the following Markov property.

3. Kauermann (1996) actually used undirected graphs with dashed edges which are Markov equivalent to graphs with only bi-directed edges (see Richardson, 2003, for discussions).

Definition 14 (The Ordered Reduced Local Markov Property (RLMP,$\prec_c$)) *Let G be an ADMG without directed mixed cycles and $\prec_c$ be a c-ordering on the vertices of G. A probability distribution P is said to satisfy the ordered reduced local Markov property for G with respect to $\prec_c$ if*

$$\text{(RLMP,}\prec_c\text{)} \quad \forall X \in V,\ I(\{X\}, \text{pa}_G(X), \text{pre}_{G,\prec_c}(X) \setminus (\{X\} \cup \text{pa}_G(X) \cup \text{sp}_G(X))). \quad (10)$$

The ordered reduced local Markov property states that *a variable is independent of its predecessors, excluding its spouses, in a c-ordering given its parents*. We now establish the equivalence of (GMP) and (RLMP,$\prec_c$).

Theorem 15 *If a probability distribution P satisfies the composition axiom and an ADMG G has no directed mixed cycles, then for a c-ordering $\prec_c$ on the vertices of G,*

$$\text{(GMP)} \iff \text{(RLMP,}\prec_c\text{)}.$$

Proof: (GMP) $\Longrightarrow$ (RLMP,$\prec_c$)
The set $\text{pre}_{G,\prec_c}(X)$ does not include any descendant of $\text{dis}_G(X)$ since $\prec_c$ is a c-ordering. We have

$$\begin{aligned}
&\text{pre}_{G,\prec_c}(X) \setminus (\{X\} \cup \text{pa}_G(X) \cup \text{sp}_G(X)) \\
&= \text{pre}_{G,\prec_c}(X) \setminus \Big(\Big(\{X\} \cup \text{pa}_G(X) \cup \text{sp}_G(X) \Big) \\
&\qquad\qquad \cup \Big(\text{de}_G(\{X\} \cup \text{sp}_G(X)) \setminus (\{X\} \cup \text{sp}_G(X)) \Big) \Big) \\
&= \text{pre}_{G,\prec_c}(X) \setminus \text{f}(X, G) \\
&\subseteq V \setminus \text{f}(X, G).
\end{aligned}$$

Hence, (RLMP,$\prec_c$) follows from (RLMP). ■

Proof: (RLMP,$\prec_c$) $\Longrightarrow$ (GMP)
We will show that (RLMP,$\prec_c$) $\Longrightarrow$ (LMP,$\prec_c$). Assume that a probability distribution P satisfies (RLMP,$\prec_c$). Let $\text{g}(Y) = \text{pre}_{G,\prec_c}(Y) \setminus (\{Y\} \cup \text{pa}_G(Y) \cup \text{sp}_G(Y))$. Consider the set of conditional independence relations invoked by (LMP,$\prec_c$) for each variable X given in (4) where A is maximal. By (10), for all Y in $\text{dis}_{G_A}(X)$, we have

$$I(Y, \text{pa}_G(Y), \text{g}(Y)). \quad (11)$$

Let $S_1 = \text{pa}_G(\text{dis}_{G_A}(X)) \setminus \text{pa}_G(Y)$ and $S_2 = A \setminus (\text{mb}(X, A) \cup \{X\})$. We have that

$$S_1 \subseteq \text{g}(Y).$$

Note that $S_2 \setminus \text{g}(Y)$ may be non-empty. Let $S_3 = S_2 \setminus \text{g}(Y)$. It suffices to show that

$$I(Y, \text{pa}_G(Y), S_3),$$

which implies $I(Y, \text{pa}_G(Y), S_2)$ by composition. Then, the rest of the proof would be identical to that of Theorem 11.

We first characterize the vertices in S_3. We will show that

$$S_3 = (\text{pre}_{G,\prec_c}(X) \setminus \text{pre}_{G,\prec_c}(Y)) \setminus \text{sp}_G(\text{dis}_{G_A}(X)). \quad (12)$$

By Lemma 6, we have

$$S_2 = \mathrm{pre}_{G,\prec_c}(X) \setminus \Big(\mathrm{de}_G(\mathrm{h}(X,A)) \cup \mathrm{mb}(X,A) \cup \{X\}\Big).$$

Since $\prec_c$ is a c-ordering, no descendant of $\mathrm{dis}_G(X)$ will appear in A. Hence,

$$S_2 = \mathrm{pre}_{G,\prec_c}(X) \setminus \Big(\mathrm{sp}_G(\mathrm{dis}_{G_A}(X)) \cup \mathrm{pa}_G(\mathrm{dis}_{G_A}(X))\Big).$$

To identify some common elements of S_2 and $\mathrm{g}(Y)$, we will reformulate S_2 and $\mathrm{g}(Y)$ as follows.

$$\begin{aligned} S_2 &= \Big(B \setminus \mathrm{pa}_G(\mathrm{dis}_{G_A}(X))\Big) \cup \Big((\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(X)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X))\Big), \\ \mathrm{g}(Y) &= \Big(B \setminus \mathrm{pa}_G(Y)\Big) \cup \Big((\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(Y)) \setminus (\{Y\} \cup \mathrm{sp}_G(Y))\Big) \end{aligned}$$

where $B = \mathrm{pre}_{G,\prec_c}(X) \setminus \mathrm{dis}_G(X)$. This can be verified by noting that $A_1 = A_2 \setminus (A_3 \cup A_4) = (A_{11} \setminus A_2) \cup (A_{12} \setminus A_3)$ if $A_1 = A_{11} \cup A_{12}$, $A_{11} \cap A_{12} = \emptyset$, $A_2 \subseteq A_{11}$, $A_3 \subseteq A_{12}$. From $\mathrm{pa}_G(Y) \subseteq \mathrm{pa}_G(\mathrm{dis}_{G_A}(X))$, it follows that $B \setminus \mathrm{pa}_G(\mathrm{dis}_{G_A}(X)) \subseteq B \setminus \mathrm{pa}_G(Y)$ and

$$\begin{aligned} S_3 =& S_2 \setminus \mathrm{g}(Y) \\ =& \Big((\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(X)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X))\Big) \\ & \setminus \Big((\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(Y)) \setminus (\{Y\} \cup \mathrm{sp}_G(Y))\Big). \end{aligned}$$

We can rewrite the first part of this expression as follows.

$$\begin{aligned} &(\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(X)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X)) \\ &= \Big((\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(Y)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X))\Big) \\ &\quad \cup \Big((\mathrm{pre}_{G,\prec_c}(X) \setminus \mathrm{pre}_{G,\prec_c}(Y)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X))\Big). \end{aligned}$$

From $(\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(Y)) \setminus \mathrm{sp}_G(\mathrm{dis}_{G_A}(X)) \subseteq (\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec_c}(Y)) \setminus (\{Y\} \cup \mathrm{sp}_G(Y))$, (12) follows. Thus, the vertices in S_3 are those in the set $\mathrm{pre}_{G,\prec_c}(X) \setminus \mathrm{pre}_{G,\prec_c}(Y)$ and not in the set $\mathrm{sp}_G(\mathrm{dis}_{G_A}(X))$.

Now we are ready to prove $I(Y, \mathrm{pa}_G(Y), S_3)$. For any $Z \in S_3$, we have $Y \prec Z$ and $Z \notin \mathrm{sp}_G(Y)$. Hence,

$I(\{Z\}, \mathrm{pa}_G(Z), \mathrm{g}(Z))$,	
$I(\{Z\}, \mathrm{pa}_G(Z), \{Y\} \cup (\mathrm{pa}_G(Y) \setminus \mathrm{pa}_G(Z)))$	by decomposition,
$I(\{Z\}, \mathrm{pa}_G(Z) \cup \mathrm{pa}_G(Y), \{Y\})$	by weak union,
$I(\{Y\}, \mathrm{pa}_G(Y), \mathrm{pa}_G(Z) \setminus \mathrm{pa}_G(Y))$ and decomposition,	by $\mathrm{pa}_G(Z) \setminus \mathrm{pa}_G(Y)) \subseteq \mathrm{g}(Y)$, (11)
$I(\{Y\}, \mathrm{pa}_G(Y), \{Z\})$	by contraction and decomposition.

Therefore, by composition, $I(Y, \mathrm{pa}_G(Y), S_3)$ holds. ■

(RLMP,$\prec_c$) invokes one zero partial correlation for each pair of nonadjacent variables. For example, for the ADMG G in Figure 3 (a) and a c-ordering $\prec_c$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec$

$V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$, (RLMP,$\prec_c$) invokes the following conditional independence relations:

$$
\begin{aligned}
&I(\{V_2\}, \emptyset, \{V_1\}), && I(\{V_3\}, \{V_1\}, \{V_2\}),\\
&I(\{V_4\}, \{V_2\}, \{V_1, V_3\}), && I(\{V_5\}, \{V_3\}, \{V_1, V_2, V_4\}),\\
&I(\{V_6\}, \{V_4\}, \{V_1, V_2, V_3\}), && I(\{V_7\}, \{V_4\}, \{V_1, V_2, V_3, V_5\}),\\
&I(\{V_8\}, \{V_6\}, \{V_1, V_2, V_3, V_4, V_5, V_7\}), && I(\{V_9\}, \{V_2, V_7\}, \{V_1, V_3, V_4, V_5, V_6\})
\end{aligned} \tag{13}
$$

which involve 25 zero partial correlations while (7) involve 37 zero partial correlations.

3.3. The Pairwise Markov Property

In this section, we give a pairwise Markov property which specifies conditional independence relations between pairs of variables and show that it is equivalent to the global Markov property. In previous sections, we focused on minimizing the number of zero partial correlations. We now take into account the size of the conditioning set Z in each zero partial correlation $\rho_{XY.Z}$. When the size of $\mathrm{pa}_G(X)$ for a vertex X in (RLMP,$\prec_c$) is large, it might be advantageous to use a different conditioning set with smaller size (if the equivalence of the Markov properties still holds). Pearl and Meshkat (1999) introduced a pairwise Markov property for DAGs (without bi-directed edges) which may involve fewer conditioning variables and thus lead to more economical tests. The result can be easily generalized to ADMGs with no directed mixed cycles.

Let $d(X, Y)$ denote the shortest distance between two vertices X and Y, that is, the number of edges in the shortest path between X and Y. Two vertices X and Y are nonadjacent if X and Y are not connected by a directed nor a bi-directed edge.

Definition 16 (The Pairwise Markov Property (PMP,$\prec_c$)) *Let G be an ADMG without directed mixed cycles and $\prec_c$ be a c-ordering on the vertices of G. A probability distribution P is said to satisfy the pairwise Markov property for G with respect to $\prec_c$ if for any two nonadjacent vertices V_i, V_j, $V_j \prec_c V_i$*

$$
\text{(PMP,}\prec_c\text{)} \qquad\qquad I(\{V_i\}, Z_{ij}, \{V_j\})
$$

where Z_{ij} is any set of vertices such that Z_{ij} d-separates V_i from V_j and $\forall Z \in Z_{ij}, d(V_i, Z) < d(V_i, V_j)$.

Note that, in ADMGs with no directed mixed cycles, there always exists such a Z_{ij} for any two nonadjacent vertices. For example, the parent set of V_i always satisfies the condition for Z_{ij}. If the empty set d-separates V_i from V_j, then the empty set is defined to satisfy the condition for Z_{ij}. Therefore we can always choose a Z_{ij} with the smallest size, providing a more economical way to test zero partial correlations.

Theorem 17 *If a probability distribution P satisfies the composition axiom and an ADMG G has no directed mixed cycles, then*

$$
\text{(GMP)} \iff \text{(PMP,}\prec_c\text{)}.
$$

Proof: Noting that two vertices X and Y are adjacent if $X \leftarrow Y$, $X \rightarrow Y$ or $X \leftrightarrow Y$, the proof of Theorem 1 by Pearl and Meshkat (1999) is directly applicable to ADMGs and it effectively proves that (RLMP,$\prec_c$) $\iff$ (PMP,$\prec_c$). We do not reproduce the proof here. ■

As an example, for the ADMG G in Figure 3 (a) and a c-ordering $\prec_c$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$, the following conditional independence relations (for convenience, we combine the relations for each vertex that have the same conditioning set) can be given by (PMP,$\prec_c$):

$$
\begin{aligned}
&I(\{V_2\},\emptyset,\{V_1\}), && I(\{V_3\},\emptyset,\{V_2\}),\\
&I(\{V_4\},\emptyset,\{V_3,V_1\}), && I(\{V_5\},\emptyset,\{V_4,V_2\}),\\
&I(\{V_5\},\{V_3\},\{V_1\}), && I(\{V_6\},\emptyset,\{V_3,V_1\}),\\
&I(\{V_6\},\{V_4\},\{V_2\}), && I(\{V_7\},\emptyset,\{V_5,V_3,V_1\}),\\
&I(\{V_7\},\{V_4\},\{V_2\}), && I(\{V_8\},\{V_6\},\{V_7,V_5,V_4,V_2\}),\\
&I(\{V_8\},\emptyset,\{V_3,V_1\}), && I(\{V_9\},\{V_2,V_7\},\{V_6,V_4\}),\\
&I(\{V_9\},\emptyset,\{V_5,V_3,V_1\}) &&
\end{aligned}
$$

which involve the same number of zero partial correlations as (13) but involve smaller conditioning sets than those in (13).

3.4. Relation to Other Work

In this section, we contrast the class of ADMGs without directed mixed cycles to maximal ancestral graphs and chain graphs in terms of Markov properties.

3.4.1. MAXIMAL ANCESTRAL GRAPHS

It is easy to see that an ADMG without directed mixed cycles is a *maximal ancestral graph (MAG)* (Richardson and Spirtes, 2002). An ADMG is said to be *ancestral* if, for any edge $X \leftrightarrow Y$, X is not an ancestor of Y (and vice versa). Note that an edge $X \leftrightarrow Y$ and a directed path from X to Y (or Y to X) form a directed mixed cycle. Hence, an ADMG without directed mixed cycles is ancestral. An ancestral graph is said to be *maximal* if, for any pair of nonadjacent vertices X and Y, there exists a set $Z \subseteq V \setminus \{X, Y\}$ that d-separates X from Y. From Theorem 17, it follows that an ADMG without directed mixed cycles is maximal. On the other hand, there exist MAGs which have directed mixed cycles (see Figure 4). Thus, the class of ADMGs without directed mixed cycles is a strict subclass of MAGs.

Richardson and Spirtes (2002, p.979) showed the following pairwise Markov property for a MAG G:

$$I(\{V_i\}, \mathrm{an}_G(\{V_i, V_j\}) \setminus \{V_i, V_j\}, \{V_j\})$$

for any two nonadjacent vertices V_i and V_j. Richardson and Spirtes (2002) proved that this pairwise Markov property implies the global Markov property assuming a Gaussian parametrization. This does not trivially imply our results in Section 3.3 and our results cannot be considered as a special case of the results on MAGs. The two pairwise Markov properties involve two different forms of conditioning sets. The pairwise Markov property for MAGs involves considerably larger conditioning sets than our pairwise Markov property: the conditioning set includes all ancestors of V_i and V_j, which is undesirable for our purpose of using the zero partial correlations to test a model.

Also, it should be stressed that our results do not depend on a specific parameterization. We only require the composition axiom to be satisfied. In contrast, Richardson and Spirtes (2002) consider only Gaussian parameterizations. It requires further study

whether the pairwise Markov property for MAGs can be generalized to the class of distributions satisfying the composition axiom.

In the next section, we consider general ADMGs and try to eliminate redundant conditional independence relations from (LMP,$\prec$). The class of MAGs is clearly a (strict) subclass of ADMGs. Hence, given a MAG, we have two options: either we use the result in the next section or the pairwise Markov property for MAGs. Although the pairwise Markov property for MAGs gives fewer zero partial correlations (one for each nonadjacent pair of vertices), it is possible that in some cases we are better off using the result in the next section (because of the cost incurred by the large conditioning sets in the pairwise Markov property for MAGs). An example of this situation will be given in the next section.

Richardson and Spirtes (2002) also proved that for a Gaussian distribution encoded by a MAG all the constraints on the distribution (that is, on the covariance matrix) are implied by the vanishing partial correlations given by the global Markov property. Hence, this also holds in a linear SEM represented by an ADMG without directed mixed cycles which is a special type of MAG.

3.4.2. CHAIN GRAPHS

The graph that results from replacing bi-directed edges with undirected edges in an ADMG without directed mixed cycles is a *chain graph*. The class of chain graphs has been studied extensively (see Lauritzen, 1996, for a review).

Some Markov properties have been proposed for chain graphs. The first Markov property for chain graphs has been proposed by Lauritzen and Wermuth (1989) and Frydenberg (1990). Andersson et al. (2001) have introduced another Markov property. These two Markov properties do not correspond to the Markov property for ADMGs. Let G be an ADMG without directed mixed cycles and G' be the chain graph obtained by replacing bi-directed edges with undirected edges. In general, the set of conditional independence relations given by the Markov property for G is not equivalent to that given by either of the two Markov properties for chain graphs. However, there are other Markov properties for chain graphs that correspond to the Markov property for ADMGs without directed mixed cycles (Cox and Wermuth, 1993; Wermuth and Cox, 2001, 2004).[4]

4. Markov Properties for General ADMGs

When an ADMG G has directed mixed cycles, (RLMP), (RLMP,$\prec_c$), and (PMP,$\prec_c$) are no longer equivalent to (GMP) while (LMP,$\prec$) still is. In this section, we show that the number of conditional independence relations given by (LMP,$\prec$) for an arbitrary ADMG that might have directed mixed cycles can still be reduced. We introduce a procedure to reduce (LMP,$\prec$). We then give an example to illustrate the procedure.

4.1. Reducing the Ordered Local Markov Property

First, we introduce a lemma that gives a condition by which a conditional independence relation renders another conditional independence relation redundant.

4. In their terminology, ADMGs without directed mixed cycles correspond to chain graphs with dashed arrows and dashed edges.

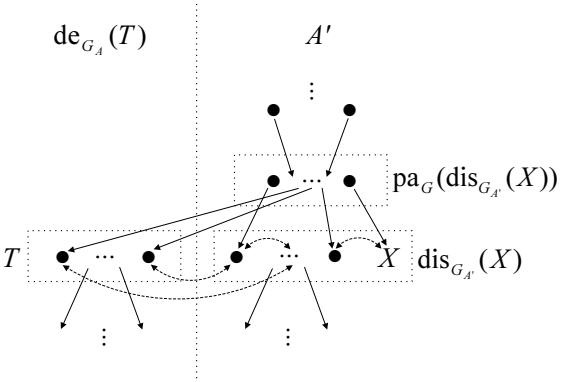


Figure 5: The relationship between A and A' that satisfy the conditions in Lemma 18. The induced subgraph G_A is shown. The vertices of G_A are decomposed into two disjoint subsets $\mathrm{de}_{G_A}(T)$ and A'.

Lemma 18 *Given an ADMG G, a consistent ordering $\prec$ on the vertices of G and a vertex X, assume that a probability distribution P satisfies the global Markov property for $G_{\mathrm{pre}_{G,\prec}(X)\setminus\{X\}}$. Let $A = \mathrm{pre}_{G,\prec}(X)$ and A' be a maximal ancestral set with respect to $\mathrm{mb}(X, A')$ such that $X \in A' \subset A$, $A' \cap \mathrm{dis}_{G_A}(X) = \mathrm{dis}_{G_{A'}}(X)$ and $\mathrm{pa}_G(\mathrm{dis}_{G_A}(X) \setminus \mathrm{dis}_{G_{A'}}(X)) \subseteq \mathrm{mb}(X, A')$. Then,*

$$I(\{X\}, \mathrm{mb}(X, A), A \setminus (\mathrm{mb}(X, A) \cup \{X\})) \tag{14}$$

implies

$$I(\{X\}, \mathrm{mb}(X, A'), A' \setminus (\mathrm{mb}(X, A') \cup \{X\})).$$

We define $\mathrm{rd}_{G,\prec}(X)$ to be the set of all A' satisfying this condition.

Proof: First, we show the relationships among $A, \mathrm{dis}_{G_A}(X), \mathrm{mb}(X, A)$ and $A', \mathrm{dis}_{G_{A'}}(X), \mathrm{mb}(X, A')$. By Lemma 6, we have

$$A' = A \setminus \mathrm{de}_{G_A}(\mathrm{h}(X, A')) \tag{15}$$

where

$$\mathrm{h}(X, A') \equiv \mathrm{sp}_{G_A}\Big(\mathrm{dis}_{G_{A'}}(X)\Big) \setminus \Big(\{X\} \cup \mathrm{mb}(X, A')\Big).$$

$\mathrm{dis}_{G_{A'}}(X)$ and $\mathrm{h}(X, A')$ are subsets of $\mathrm{dis}_{G_A}(X)$. Since $\mathrm{dis}_{G_{A'}}(X) \subseteq \{X\} \cup \mathrm{mb}(X, A')$ (by the definition of the Markov blanket), $\mathrm{dis}_{G_{A'}}(X) \cap \mathrm{h}(X, A') = \emptyset$. Thus, we can decompose the set $\mathrm{dis}_{G_A}(X)$ into 3 disjoint subsets as follows.

$$\mathrm{dis}_{G_A}(X) = \mathrm{dis}_{G_{A'}}(X) \cup \mathrm{h}(X, A') \cup B \tag{16}$$

where

$$B \equiv \mathrm{dis}_{G_A}(X) \setminus \Big(\mathrm{dis}_{G_{A'}}(X) \cup \mathrm{h}(X, A')\Big).$$

We have

$$\begin{aligned} A' \cap \mathrm{dis}_{G_A}(X) &= A' \cap \Big(\mathrm{dis}_{G_{A'}}(X) \cup \mathrm{h}(X, A') \cup B\Big) \\ &= \mathrm{dis}_{G_{A'}}(X) \cup B \end{aligned}$$

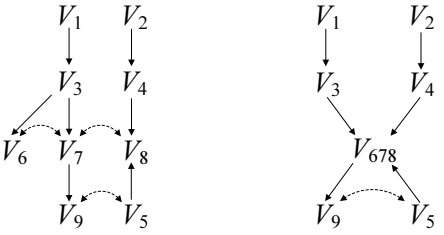


Figure 6: (a) An ADMG with directed mixed cycles (b) Illustration of the procedure **GetOrdering**. The modified graph after the first step is shown.

since $\text{dis}_{G_{A'}}(X) \subseteq A', B \subseteq A'$ and $A' \cap \text{h}(X, A') = \emptyset$. From the assumption in Lemma 18 that $A' \cap \text{dis}_{G_A}(X) = \text{dis}_{G_{A'}}(X)$, it follows that $B = \emptyset$. Thus, from (16), we have

$$\text{dis}_{G_A}(X) \setminus \text{dis}_{G_{A'}}(X) = \text{h}(X, A'). \tag{17}$$

Let $T = \text{dis}_{G_A}(X) \setminus \text{dis}_{G_{A'}}(X) = \text{h}(X, A')$. Then,

$$\begin{aligned} \text{mb}(X, A) &= \text{mb}(X, A') \cup T \cup \text{pa}_G(T) \\ &= \text{mb}(X, A') \cup T \end{aligned} \tag{18}$$

since $\text{pa}_G(T) \subseteq \text{mb}(X, A')$ by our assumption. Thus A decomposes into

$$A = A' \cup \text{de}_{G_A}(T) \tag{19}$$

since $\text{de}_{G_A}(T) \subseteq A$ and (15).

The key relationships among $A, \text{dis}_{G_A}(X), \text{mb}(X, A)$ and $A', \text{dis}_{G_{A'}}(X), \text{mb}(X, A')$ are given by (17)–(19). Figure 5 shows these relationships. We are now ready to prove that $I(\{X\}, \text{mb}(X, A'), A' \setminus (\text{mb}(X, A') \cup \{X\}))$ can be derived from $I(\{X\}, \text{mb}(X, A), A \setminus (\text{mb}(X, A) \cup \{X\}))$. From (18) and (19), it follows that

$$A \setminus (\text{mb}(X, A) \cup \{X\}) = (A' \cup \text{de}_{G_A}(T)) \setminus (\text{mb}(X, A') \cup \{X\} \cup T).$$

Since $A' \cap \text{de}_{G_A}(T) = \emptyset, (\text{mb}(X, A') \cup \{X\}) \cap T = \emptyset, \text{mb}(X, A') \cup \{X\} \subseteq A'$ and $T \subseteq \text{de}_{G_A}(T)$, we have

$$A \setminus (\text{mb}(X, A) \cup \{X\}) = \Big(A' \setminus (\text{mb}(X, A') \cup \{X\})\Big) \cup \Big(\text{de}_{G_A}(T) \setminus T\Big). \tag{20}$$

Plugging (18) and (20) into (14), we get

$$I\Big(\{X\}, \text{mb}(X, A') \cup T, \Big(A' \setminus (\text{mb}(X, A') \cup \{X\})\Big) \cup \Big(\text{de}_{G_A}(T) \setminus T\Big)\Big).$$

From the decomposition axiom, it follows that

$$I(\{X\}, \text{mb}(X, A') \cup T, A' \setminus (\text{mb}(X, A') \cup \{X\})). \tag{21}$$

The last step is to remove T from the conditioning set to obtain $I(\{X\}, \mathrm{mb}(X, A'), A' \setminus (\mathrm{mb}(X, A') \cup \{X\}))$. We claim that

$$I(T, \mathrm{mb}(X, A'), A' \setminus (\mathrm{mb}(X, A') \cup \{X\})). \tag{22}$$

We first argue that T is d-separated from $A' \setminus (\mathrm{mb}(X, A') \cup \{X\})$ given $\mathrm{mb}(X, A')$. Consider a vertex $t \in T$ and a vertex $\alpha \in A' \setminus (\mathrm{mb}(X, A') \cup \{X\})$. Note that for any bi-directed edge $t \leftrightarrow \beta$ in G_A, β is either in T or $\mathrm{dis}_{G_{A'}}(X)$. There are only four possible cases for any path in G_A from t to α.

1. $t \leftarrow \gamma \cdots \alpha$
2. $t \rightarrow \cdots \rightarrow \gamma \leftarrow * \cdots \alpha$
3. $t \leftrightarrow \leftrightarrow \cdots \leftrightarrow \delta \leftarrow \gamma \cdots \alpha$
4. $t \leftrightarrow \leftrightarrow \cdots \leftrightarrow \delta \rightarrow \cdots \rightarrow \gamma \leftarrow * \cdots \alpha$

In case 1, $\gamma \in \mathrm{mb}(X, A')$ since $\mathrm{pa}_G(T) \subseteq \mathrm{mb}(X, A')$. Thus, the path is not d-connecting. In case 2, γ is a descendant of t. Since $\mathrm{mb}(X, A')$ does not contain any descendant of t, the path is not d-connecting. Case 3 is similar to case 1, but there are one or more bi-directed edges after t. δ is either in T or $\mathrm{dis}_{G_{A'}}(X)$. It follows that $\gamma \in \mathrm{mb}(X, A')$, so the path is not d-connecting. Case 4 is similar to case 2, but there are one or more bi-directed edges after t. If δ is in T, the argument for case 2 can be applied. If δ is in $\mathrm{dis}_{G_{A'}}(X)$, then $\delta \in \mathrm{mb}(X, A')$, which implies that the path is not d-connecting. This establishes that T is d-separated from $A' \setminus (\mathrm{mb}(X, A') \cup \{X\})$ given $\mathrm{mb}(X, A')$. By the assumption that P satisfies the global Markov property for $G_{\mathrm{pre}_{G,\prec}(X)\setminus\{X\}}$, (22) holds. Finally, from (21),(22) and the contraction axiom, it follows that $I(\{X\}, \mathrm{mb}(X, A'), A' \setminus (\mathrm{mb}(X, A') \cup \{X\}))$. ■

For example, consider the ADMG G in Figure 2 and a consistent ordering $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7$. Assume that the global Markov property for $G_{\mathrm{pre}_{G,\prec}(V_6)}$ is satisfied. Let $A = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$ and $A' = \{V_1, V_2, V_3, V_4, V_6, V_7\}$. Then, $\mathrm{dis}_{G_A}(V_7) = \{V_5, V_6, V_7\}$, $\mathrm{dis}_{G_{A'}}(V_7) = \{V_6, V_7\}$, $A' \cap \mathrm{dis}_{G_A}(V_7) = \{V_6, V_7\} = \mathrm{dis}_{G_{A'}}(V_7)$ and $\mathrm{pa}_G(\mathrm{dis}_{G_A}(V_7) \setminus \mathrm{dis}_{G_{A'}}(V_7)) = \{V_3\} \subseteq \{V_3, V_4, V_6\} = \mathrm{mb}(V_7, A')$. Thus, $I(\{V_7\}, \{V_3, V_4, V_6\}, \{V_1, V_2\})$ follows from $I(\{V_7\}, \{V_3, V_4, V_5, V_6\}, \{V_1, V_2\})$. Note that in the proof of Lemma 18, the composition axiom is not used. Thus, Lemma 18 can be used to reduce the ordered local Markov property for ADMGs associated with an arbitrary probability distribution. Also, note that the condition that P satisfies the global Markov property for $G_{\mathrm{pre}_{G,\prec}(X)\setminus\{X\}}$ is always satisfied in a recursive application of this lemma in Theorem 21.

We now introduce a key concept in eliminating redundant conditional independence relations from (LMP,$\prec$).

Definition 19 (C-ordered Vertex) *Given a consistent ordering $\prec$ on the vertices of an ADMG G, a vertex X is said to be c-ordered in $\prec$ if*

1. *all vertices in $\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X)$ are consecutive in $\prec$ and*
2. *for any two vertices Y and Z in $\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X)$, there is no directed edge between Y and Z.*

procedure ReduceMarkov
INPUT: An ADMG G and a consistent ordering $\prec$ on the vertices of G
OUTPUT: A set of conditional independence relations S
$S \leftarrow \emptyset$
for $i = 1, \ldots, n$ **do**
  $I_i \leftarrow \emptyset$
  if V_i is c-ordered in $\prec$ **then**
    for nonadjacent $V_j \prec V_i$ **do**
      $I_i \leftarrow I_i \cup I(\{V_i\}, Z_{ij}, \{V_j\})$ where Z_{ij} is any set of vertices such that Z_{ij} d-separates V_i from V_j and $\forall Z \in Z_{ij}, d(V_i, Z) < d(V_i, V_j)$
    end for
  else
    for all maximal ancestral sets A with respect to $\mathrm{mb}(V_i, A)$ such that $V_i \in A \subseteq \mathrm{pre}_{G,\prec}(V_i)$, $A \notin \mathrm{rd}_{G,\prec}(V_i)$ **do**
      $I_i \leftarrow I_i \cup I(\{V_i\}, \mathrm{mb}(V_i, A), A \setminus (\mathrm{mb}(V_i, A) \cup \{V_i\}))$
    end for
  end if
  $S \leftarrow S \cup I_i$
end for

Figure 7: A procedure to generate a reduced set of conditional independence relations for an ADMG G and a consistent ordering $\prec$

If no bi-directed edge is connected to X, then X is defined to be c-ordered. For example, consider the ADMG G in Figure 6 (a). $\prec$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$ is a consistent ordering on the vertices of G. $V_1, V_2, \ldots, V_8$ are c-ordered in $\prec$ but V_9 is not since V_5 and V_9 are not consecutive in $\prec$.

The key observation, which will be proved, is that c-ordered vertices contribute to eliminating many redundant conditional independence relations invoked by the ordered local Markov property (LMP,$\prec$). We provide two procedures. The first procedure **ReduceMarkov** in Figure 7 constructs a list of conditional independence relations in which some redundant conditional independence relations from (LMP,$\prec$) are not included (all the conditional independence relations identified by Lemma 18 are not included). **ReduceMarkov** takes as input a fixed ordering $\prec$. The second procedure **GetOrdering** in Figure 9 gives a good ordering that might have many c-ordered vertices.

We first describe the procedure **ReduceMarkov**. Given an ADMG G and a consistent ordering $\prec$, **ReduceMarkov** gives a set of conditional independence relations which will be shown to be equivalent to the global Markov property for G. For each vertex V_i, **ReduceMarkov** generates a set of conditional independence relations. If V_i is c-ordered, the relations that correspond to the pairwise Markov property are generated. Otherwise, the relations that correspond to the ordered local Markov property are generated, and Lemma 18 is used to remove some redundant relations. The output

$S = \mathbf{ReduceMarkov}(G, \prec)$ can be described as follows:

$$S = \bigcup_{X:X \text{ is c-ordered in } \prec} \Big(\bigcup_{Y:Y \prec X} I(\{X\}, Z_{XY}, \{Y\}) \Big) \bigcup \bigcup_{X:X \text{ is not c-ordered in } \prec} \Big(\bigcup_{\substack{\text{all maximal} \\ \text{sets } A \text{ with respect} \\ \text{to mb}(X,A): \\ X \in A \subseteq \text{pre}_{G,\prec}(X), \\ A \notin \text{rd}_{G,\prec}(X)}} I(\{X\}, \text{mb}(X, A), A \setminus (\text{mb}(X, A) \cup \{X\})) \Big) \tag{23}$$

where Z_{XY} is any set of vertices such that Z_{XY} d-separates X from Y and $\forall Z \in Z_{XY}$, $d(X, Z) < d(X, Y)$.

If a vertex X is c-ordered, $O(n)$ conditional independence relations (or zero partial correlations) are added to S. Otherwise, $O(2^n)$ conditional independence relations may be added to S and $O(n2^n)$ zero partial correlations may be invoked. Furthermore, a c-ordered vertex typically involves a smaller conditioning set. $I(\{X\}, Z_{XY}, \{Y\})$ has the conditioning set $|Z_{XY}| \leq |\text{pa}_G(X)|$ while $I(\{X\}, \text{mb}(X, A), A \setminus (\text{mb}(X, A) \cup \{X\}))$ has the conditioning set $|\text{mb}(X, A)| \geq |\text{pa}_G(X)|$.

We now prove that the conditional independence relations produced by **ReduceMarkov** can derive all the conditional independence relations invoked by the global Markov property.

Definition 20 (*S*-Markov Property (*S*-MP,$\prec$)) *Let G be an ADMG and $\prec$ be a consistent ordering on the vertices of G. Let S be the set of conditional independence relations given by **ReduceMarkov**$(G,\prec)$. A probability distribution P is said to satisfy the S-Markov property for G with respect to $\prec$, if*

(S-MP,$\prec$) *P satisfies all the conditional independence relations in S.*

Theorem 21 *Let G be an ADMG and $\prec$ be a consistent ordering on the vertices of G. Let S be the set of conditional independence relations given by **ReduceMarkov**$(G,\prec)$. If a probability distribution P satisfies the composition axiom, then*

$$\text{(GMP)} \iff (S\text{-MP},\prec).$$

Proof: (GMP) $\Longrightarrow$ (S-MP,$\prec$) since every conditional independence relation in (S-MP,$\prec$) corresponds to a valid d-separation. We show (S-MP,$\prec$) $\Longrightarrow$ (GMP). Without any loss of generality, let $\prec$: $V_1 \prec \ldots \prec V_n$. The proof is by induction on the sequence of ordered vertices. Suppose that (S-MP,$\prec$) $\Longrightarrow$ (GMP) holds for $V_1, \ldots V_{i-1}$. Let $S_{i-1} = I_1 \cup \ldots \cup I_{i-1}$. Then, by the induction hypothesis, S_{i-1} contains all the conditional independence relations invoked by (LMP,$\prec$) for $V_1, \ldots V_{i-1}$. If V_i is not c-ordered, $I_i = I(\{V_i\}, \text{mb}(V_i, A), A \setminus (\text{mb}(V_i, A) \cup \{V_i\}))$ for all maximal ancestral sets A such that $V_i \in A \subseteq \text{pre}_{G,\prec}(V_i)$, $A \notin \text{rd}_{G,\prec}(V_i)$. The conditional independence relations invoked by (LMP,$\prec$) with respect to V_i and any $A \in \text{rd}_{G,\prec}(V_i)$ can be derived from other conditional independence relations by Lemma 18. Thus, $S_i = S_{i-1} \cup I_i$ contains all the conditional independence relations invoked by (LMP,$\prec$) for $V_1, \ldots V_i$, which implies (GMP). If V_i is c-ordered, applying the arguments in the proof of (GMP) $\iff$ (PMP,$\prec_c$), we have

$$I(\{V_i\}, \text{pa}_G(V_i), \text{pre}_{G,\prec}(V_i) \setminus (\{V_i\} \cup \text{pa}_G(V_i) \cup \text{sp}_G(V_i))).$$

By the induction hypothesis and the definition of a c-ordered vertex, we have for all $V_j \in \mathrm{dis}_G(V_i) \cap \mathrm{pre}_{G,\prec}(V_i)$

$$I(\{V_j\}, \mathrm{pa}_G(V_j), \mathrm{pre}_{G,\prec}(V_j) \setminus (\{V_j\} \cup \mathrm{pa}_G(V_j) \cup \mathrm{sp}_G(V_j))).$$

By the arguments in the proof of (GMP) $\iff$ (RLMP,$\prec_c$), we have for all maximal ancestral sets A such that $V_i \in A \subseteq \mathrm{pre}_{G,\prec}(V_i)$

$$I(\{V_i\}, \mathrm{mb}(V_i, A), A \setminus (\mathrm{mb}(V_i, A) \cup \{V_i\})).$$

Therefore, $S_i = S_{i-1} \cup I_i$ derives all the conditional independence relations invoked by (GMP). ■

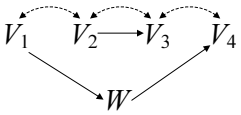


Figure 8: The c-component $\{V_1, V_2, V_3, V_4\}$ has the root set $\{V_1, V_2\}$

As we have seen earlier, the number of zero partial correlations critically depends on the number of c-ordered vertices in a given ordering. This motivates us to find the ordering with the most c-ordered vertices. An obvious way of finding this ordering is to explore the space of all the consistent orderings. However, this exhaustive search may become infeasible as the number of vertices grows. We propose a greedy algorithm to get an ordering that has a large number of c-ordered vertices. The basic idea is to first find a large c-component in which many vertices can be c-ordered and place the vertices consecutively in the ordering, then repeating this until we cannot find a set of vertices that can be c-ordered. To describe the algorithm, we define the following notion, which identifies the largest subset of a c-component that can be c-ordered.

Definition 22 (Root Set) *The root set of a c-component C, denoted* $\mathrm{rt}(C)$ *is defined to be the set* $\{V_i \in C \mid$ *there is no* $V_j \in C$ *such that a directed path* $V_j \to \ldots \to V_i$ *exists in* $G\}$.

For example, the c-component $\{V_1, V_2, V_3, V_4\}$ in Figure 8 has the root set $\{V_1, V_2\}$. V_3 and V_4 are not in the root set since there are paths $V_2 \to V_3$ and $V_1 \to W \to V_4$. The root set has the following properties.

Proposition 23 *Let $\prec$ be a consistent ordering on the vertices of an ADMG G and C be a c-component of G. If the vertices in* $\mathrm{rt}(C)$ *are consecutive in $\prec$, then all the vertices in* $\mathrm{rt}(C)$ *are c-ordered in $\prec$.*

Proof: Assume that the vertices in $\mathrm{rt}(C)$ are consecutive in $\prec$. Then, for $X \in \mathrm{rt}(C)$, $\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X) \subseteq \mathrm{rt}(C)$. Thus, there is no directed edge between any two vertices in $\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X)$. ■

Proposition 24 *Let $\prec$ be a consistent ordering on the vertices of an ADMG G and C be a c-component of G. If a vertex X in C is c-ordered in $\prec$, then $X \in \mathrm{rt}(C)$.*

Proof: Assume that X is c-ordered in $\prec$. Suppose for a contradiction that $X \notin \mathrm{rt}(C)$. Then, there exists an ancestor Y of X in C. If there exists a vertex Z such that $Z \notin C$,

procedure GetOrdering
INPUT: An ADMG G
OUTPUT: A consistent ordering $\prec$ on V
Step 1:
$G' \leftarrow G$ (V' is the set of vertices of G)
while (there is a c-component C of G' such that $|\mathrm{rt}(C)| > 1$) **do**
  $M \leftarrow \emptyset$
  for each c-component C of G' **do**
    if $|\mathrm{rt}(C)| > |M|$ **then**
      $M \leftarrow \mathrm{rt}(C)$
    end if
  end for
  Add a vertex V_M to $G'_{V' \setminus M}$
  Draw an edge $V_M \leftarrow X$ (respectively $V_M \rightarrow X$, $V_M \leftrightarrow X$) if there is
  $Y \leftarrow X$ (respectively $Y \rightarrow X$, $Y \leftrightarrow X$) in G' such that $Y \in M, X \in V' \setminus M$
  Let G' be the resulting graph
end while
Step 2:
Let $\prec'$ be any consistent ordering on V'. Construct a consistent ordering $\prec$ from $\prec'$ by replacing each $V_S \in V' \setminus V$ with the vertices in S (the ordering of the vertices in S is arbitrary)

Figure 9: A greedy algorithm to generate a good consistent ordering on the vertices of an ADMG G

$Y \rightarrow \cdots \rightarrow Z \rightarrow \cdots \rightarrow X$. Then, the first condition of a c-ordered vertex is violated. Otherwise, the second condition is violated. ■

Proposition 23 and 24 imply that the root set of a c-component is the largest subset of the c-component that can be c-ordered in a consistent ordering. If G does not have directed mixed cycles, $\mathrm{rt}(C) = C$ for every c-component C.

The procedure **GetOrdering** in Figure 9 is our proposed greedy algorithm that generates a good consistent ordering for G. In Step 1, it searches for the largest root set M and then merges all the vertices in M to one vertex V_M modifying edges accordingly. Then, it repeats the same operation for the modified graph until there is no root set that contains more than one vertex. Since the vertices in a root set are merged at each iteration, the modified graph is acyclic as otherwise there would be a directed path between two vertices in the root set, which contradicts the condition of a root set. After Step 1, we can easily obtain a consistent ordering for the original graph from the modified graph.

4.2. An Example

We show the application of the procedures **ReduceMarkov** and **GetOrdering** by considering the ADMG G in Figure 6 (a). First, we apply **GetOrdering** to get a consistent ordering on the vertices V of G. In Step 1, we first look for the largest root set. The c-component $\{V_6, V_7, V_8\}$ has the largest root set $\{V_6, V_7, V_8\}$. Then, the vertices in $\{V_6, V_7, V_8\}$ are merged into a vertex V_{678}. Figure 6 (b) shows the modified graph G' after the first iteration of the while loop. In the next iteration, we find that every c-

component has the root set of size 1. Note that for $C = \{V_5, V_9\}$, $\mathrm{rt}(C) = \{V_5, V_9\}$ in G but $\mathrm{rt}(C) = \{V_5\}$ in G'. Thus, Step 1 ends. In Step 2, from G' in Figure 6 (b), we can obtain an ordering $\prec'$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_{678} \prec V_9$. This is converted to a consistent ordering $\prec$: $V_1 \prec V_2 \prec V_3 \prec V_4 \prec V_5 \prec V_6 \prec V_7 \prec V_8 \prec V_9$ for G.

With the ordering $\prec$, we now apply **ReduceMarkov** to obtain a set of conditional independence relations that can derive those invoked by the global Markov property. It is easy to see that the vertices $V_1, \ldots, V_8$ are c-ordered in $\prec$. Thus, the following conditional independence relations corresponding to the pairwise Markov property are added to the set S (initially empty).

$$
\begin{aligned}
&I(\{V_2\}, \emptyset, \{V_1\}), && I(\{V_3\}, \emptyset, \{V_2\}),\\
&I(\{V_4\}, \emptyset, \{V_3, V_1\}), && I(\{V_5\}, \emptyset, \{V_4, V_3, V_2, V_1\}),\\
&I(\{V_6\}, \emptyset, \{V_5, V_4, V_2\}), && I(\{V_6\}, \{V_3\}, \{V_1\}),\\
&I(\{V_7\}, \emptyset, \{V_5, V_4, V_2\}), && I(\{V_7\}, \{V_3\}, \{V_1\}),\\
&I(\{V_8\}, \emptyset, \{V_6, V_3, V_1\}), && I(\{V_8\}, \{V_4\}, \{V_2\}).
\end{aligned} \tag{24}
$$

V_9 is not c-ordered in $\prec$ since V_5 is not adjacent in $\prec$. Thus, we use the ordered local Markov property (LMP,$\prec$) for V_9. The maximal ancestral sets that we need to consider are

$$
\begin{aligned}
A_1 &= \mathrm{an}_G(\{V_6, V_8, V_9\}) = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9\} \text{ and}\\
A_2 &= \mathrm{an}_G(\{V_4, V_6, V_9\}) = \{V_1, V_2, V_3, V_4, V_6, V_7, V_9\}.
\end{aligned}
$$

The corresponding conditional independence relations are

$$I(\{V_9\}, \{V_7, V_5\}, \{V_8, V_6, V_4, V_3, V_2, V_1\}), \tag{25}$$
$$I(\{V_9\}, \{V_7\}, \{V_6, V_4, V_3, V_2, V_1\}). \tag{26}$$

However, it turns out that $A_2 \in \mathrm{rd}_{G,\prec}(V_9)$ and (26) is not added to S. We check the condition of Lemma 18. The global Markov property for $G_{\mathrm{pre}_{G,\prec}(V_8)}$ is satisfied by (24). Also,

$$
\begin{aligned}
\mathrm{dis}_{G_{A_1}}(V_9) &= \{V_5, V_9\},\\
\mathrm{dis}_{G_{A_2}}(V_9) &= \{V_9\},\\
A_2 \cap \mathrm{dis}_{G_{A_1}}(V_9) &= \{V_9\} = \mathrm{dis}_{G_{A_2}}(V_9),\\
\mathrm{pa}_G(\mathrm{dis}_{G_{A_1}}(V_9) \setminus \mathrm{dis}_{G_{A_2}}(V_9)) &= \emptyset \subseteq \{V_7\} = \mathrm{mb}(V_9, A_2).
\end{aligned}
$$

Therefore, the condition of Lemma 18 is satisfied and it follows that (26) is redundant. To see how much we reduced the testing requirements, the conditional independence relations invoked by (LMP,$\prec$) are shown below.

$$
\begin{aligned}
&I(\{V_2\}, \emptyset, \{V_1\}), && I(\{V_3\}, \{V_1\}, \{V_2\}),\\
&I(\{V_4\}, \{V_2\}, \{V_3, V_1\}), && I(\{V_5\}, \emptyset, \{V_4, V_3, V_2, V_1\}),\\
&I(\{V_6\}, \{V_3\}, \{V_5, V_4, V_2, V_1\}), && I(\{V_7\}, \{V_3\}, \{V_5, V_4, V_2, V_1\}),\\
&I(\{V_7\}, \{V_6, V_3\}, \{V_5, V_4, V_2, V_1\}), && I(\{V_8\}, \{V_5, V_4\}, \{V_6, V_3, V_2, V_1\}),\\
&I(\{V_8\}, \{V_7, V_5, V_4, V_3\}, \{V_2, V_1\}), && I(\{V_8\}, \{V_7, V_6, V_5, V_4, V_3\}, \{V_2, V_1\}),\\
&I(\{V_9\}, \{V_7\}, \{V_6, V_4, V_3, V_2, V_1\}), && I(\{V_9\}, \{V_7, V_5\}, \{V_8, V_6, V_4, V_3, V_2, V_1\}).
\end{aligned} \tag{27}
$$

S invokes 26 zero partial correlations while (LMP,$\prec$) invokes 39. Also, S involves much smaller conditioning sets. We have at most one vertex in each conditioning set in (24) and two vertices in (25) while 23 zero partial correlations in (27) involve more than 2 vertices in the conditioning set.

The ADMG G in this example turns out to be a MAG. As we discussed in Section 3.4.1, we have two options: either we use the constraints in (24) and (25) or the constraints given by the pairwise Markov property for MAGs. In this example, both sets of constraints involve the same number of zero partial correlations. However, the pairwise Markov property for MAGs involves much larger conditioning sets. For example, the pairwise Markov property for MAGs gives the following conditional independence relation for the pair V_6 and V_8: $I(\{V_8\}, \{V_5, V_4, V_3, V_2, V_1\}, \{V_6\})$. Our method uses an empty set as the conditioning set for the pair. Hence, in this example, we are better off using the constraints in (24) and (25).

4.3. Comparison of (LMP,$\prec$) and (S-MP,$\prec$)

From (23), it is clear that (S-MP,$\prec$) invokes fewer conditional independence relations than (LMP,$\prec$) if there are c-ordered vertices in $\prec$. But how much more economical is (S-MP,$\prec$) than (LMP,$\prec$) and for what type of graphs is the reduction large?

For simplicity, we will compare the number of conditional independence relations rather than zero partial correlations and ignore the reduction done by Lemma 18. For now assume

$$S = \bigcup_{X:X \text{ is c-ordered in } \prec} I(\{X\}, \mathrm{pa}_G(X), \mathrm{pre}_{G,\prec}(X) \setminus (\{X\} \cup \mathrm{pa}_G(X) \cup \mathrm{sp}_G(X))) \bigcup$$
$$\bigcup_{X:X \text{ is not c-ordered in } \prec} \Big(\bigcup_{\substack{\text{all maximal sets } A \\ \text{with respect} \\ \text{to mb}(X,A): \\ X \in A \subseteq \mathrm{pre}_{G,\prec}(X)}} I(\{X\}, \mathrm{mb}(X,A), A \setminus (\mathrm{mb}(X,A) \cup \{X\})) \Big).$$

Let $\mathrm{M}(X, \prec)$ be the number of different Markov blankets of a vertex X, that is, $\mathrm{M}(X, \prec) = \Big|\{\mathrm{dis}_{G_A}(X) \mid A \text{ is an ancestral set such that } X \in A \subseteq \mathrm{pre}_{G,\prec}(X)\}\Big|$, and $\mathrm{C}(\prec)$ be the set of vertices that are c-ordered in $\prec$. Then, (LMP,$\prec$) lists $\sum_{X \in V} \mathrm{M}(X, \prec)$ conditional independence relations and (S-MP,$\prec$) lists $|\mathrm{C}(\prec)| + \sum_{X \notin \mathrm{C}(\prec)} \mathrm{M}(X, \prec)$ conditional independence relations. Hence, the difference in the number of conditional independence relations between (LMP,$\prec$) and (S-MP,$\prec$) is

$$\sum_{X \in \mathrm{C}(\prec)} \Big(\mathrm{M}(X, \prec) - 1\Big).$$

This difference is large when $|\mathrm{C}(\prec)|$ or $\mathrm{M}(X, \prec)$ for each X is large.

The size of $\mathrm{C}(\prec)$ depends on the number of directed mixed cycles. From Definition 19, it follows that $\mathrm{C}(\prec)$ is large if there are a small number of directed mixed cycles. Note that a directed mixed cycle such as that in Figure 4 induces the violation of the first condition in Definition 19 and a directed mixed cycle of the form $\alpha \overset{\leftrightarrow}{\to} \beta$ induces the violation of the second condition in Definition 19.

$\mathrm{M}(X, \prec)$ depends on the structure of $\mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X)$. We will reformulate $\mathrm{M}(X, \prec)$ to show the properties that affect $\mathrm{M}(X, \prec)$. Let $G_{\leftrightarrow,\mathrm{dis}}(X, \prec) = (V', E')$ where $V' = \mathrm{dis}_G(X) \cap \mathrm{pre}_{G,\prec}(X)$ and $E' = \{V_i \leftrightarrow V_j \mid V_i \leftrightarrow V_j \text{ in } G_{V'}\}$. For example, for an

ADMG G in Figure 8 and an ordering $V_1 \prec V_2 \prec V_3 \prec V_4$, $G_{\leftrightarrow,\text{dis}}(V_3, \prec)$ is $V_1 \leftrightarrow V_2 \leftrightarrow V_3$. Let $G_{\leftrightarrow,\text{dis}}(X, \prec)_S$ be the induced subgraph of $G_{\leftrightarrow,\text{dis}}(X, \prec)$ on a set $S \subseteq \text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)$. Then, $\text{M}(X, \prec) = \Big|\{S \mid S \subseteq \text{dis}_G(X) \cap \text{pre}_{G,\prec}(X) \text{ such that } G_{\leftrightarrow,\text{dis}}(X, \prec)_S \text{ is a } \textit{connected component} \text{ of } G_{\leftrightarrow,\text{dis}}(X, \prec)_{S \cup (\text{an}_G(S) \cap \text{dis}_G(X) \cap \text{pre}_{G,\prec}(X))}\}\Big|$ that is, $\text{M}(X, \prec)$ corresponds to a set of subsets S of $\text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)$ satisfying two conditions: (i) $G_{\leftrightarrow,\text{dis}}(X, \prec)_S$ is connected; and (ii) for all $Y \in \Big(\text{an}_G(S) \cap \text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)\Big) \setminus S$, there is no path from Y to any vertices in S. The condition (i) implies that $\text{M}(X, \prec)$ will be large if the vertices in $\text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)$ are connected by many bi-directed edges. The condition (ii) implies that $\text{M}(X, \prec)$ will be large if there are few directed mixed cycles. Note that for ADMGs without directed mixed cycles, (ii) trivially holds since $\Big(\text{an}_G(S) \cap \text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)\Big) \setminus S = \emptyset$. For example, consider a subset of vertices $\{V_1, \ldots, V_k\}$ in an ADMG with edges $V_i \leftrightarrow V_k, i = 1, \ldots, k-1$, which has no directed mixed cycles. Then, for an ordering $V_1 \prec \ldots \prec V_k$, $\text{M}(V_k, \prec) = 2^{k-1}$. Also, consider a subset of vertices $\{V_1, \ldots, V_k\}$ in an ADMG with edges $V_1 \overset{\leftrightarrow}{\rightarrow} V_2 \overset{\leftrightarrow}{\rightarrow} \cdots \overset{\leftrightarrow}{\rightarrow} V_k$, which has $k-1$ directed mixed cycles. Then, $\text{M}(V_k, \prec) = 1$. Hence, it is clear that $\text{M}(X, \prec)$ is large if

1. the set $\text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)$ is large,
2. there are many bi-directed edges connecting vertices in $\text{dis}_G(X) \cap \text{pre}_{G,\prec}(X)$, and
3. there are few directed mixed cycles.

Thus, (LMP,$\prec$) will invoke a large number of conditional independence relations for an ADMG with few directed mixed cycles and large c-components with many bi-directed edges. For such an ADMG, $\sum_{X \in \text{C}(\prec)} \Big(\text{M}(X, \prec) - 1\Big)$, the reduction made by (S-MP,$\prec$), is also large. An extreme case is an ADMG that has no directed mixed cycles and each c-component of which is a clique joined by bi-directed edges. An example of such an ADMG is given in Figure 10. For this ADMG and an ordering $W \prec V \prec X \prec Y \prec Z$, (LMP,$\prec$) invokes $\text{M}(W, \prec) + \text{M}(V, \prec) + \text{M}(X, \prec) + \text{M}(Y, \prec) + \text{M}(Z, \prec) = 1 + 1 + 1 + 2 + 4 = 9$ conditional independence relations while (S-MP,$\prec$) invokes $|\text{C}(\prec)| = n = 5$ conditional independence relations. If we enlarge the clique joined by bi-directed edges such that it contains k vertices, then (LMP,$\prec$) invokes $2 + \sum_{i=0}^{k-1} 2^i = 1 + 2^k$ conditional independence relations while (S-MP,$\prec$) invokes $k + 2$.

In general, although (S-MP,$\prec$) greatly reduces (LMP,$\prec$), it may still invoke an exponential number of conditional independence relations if there exist directed mixed cycles.

5. Conclusion and Discussion

We present local Markov properties for ADMGs representing linear SEMs with correlated errors. The results have applications in testing linear SEMs against the data by testing for zero partial correlations implied by the model. For general linear SEMs with correlated errors, we provide a procedure that lists a subset of zero partial correlations that will imply all other zero partial correlations implied by the model. In particular, for a class of models whose corresponding path diagrams contain no directed mixed cycles, this subset invokes one zero partial correlation for each pair of variables.

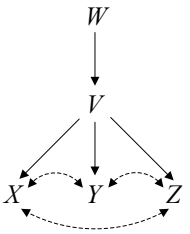


Figure 10: An example ADMG for which using (S-MP, $\prec$) is most beneficial. There is no directed mixed cycle and each c-component is a clique joined by bi-directed edges.

In general, our procedure may invoke an exponential number of zero partial correlations if the path diagram G satisfies all of the following properties: (i) G has large c-components; (ii) the vertices in each c-component are heavily connected by bi-directed edges; and (iii) G has directed mixed cycles. If one of these properties is not satisfied, then the number of zero partial correlations derived by our method is typically not exponential.

For the class of MAGs, which is a strict superclass of ADMGs without directed mixed cycles, one might use the pairwise Markov property for MAGs given in Richardson and Spirtes (2002) instead of our results in Section 4. However, when the two approaches give a similar number of constraints, it may be better to use our approach since it may use smaller conditioning sets as shown in the example in Section 4.2.

The potential advantages of testing linear SEMs based on vanishing partial correlations over the classical test method based on maximum likelihood estimation of the covariance matrix have been discussed in Pearl (1998), Shipley (2000), McDonald (2002) and Shipley (2003). The results presented in this paper provide a theoretical foundation for the practical applications of this test method in linear SEMs with correlated errors. How to implement this test method in practice still needs further study as it requires multiple testing of hypotheses about zero partial correlations (Shipley, 2000; Drton and Perlman, 2007). We also note that, in linear SEMs *without* correlated errors, all the constraints on the covariance matrix are implied by vanishing partial correlations. This also holds in linear SEMs *with* correlated errors that are represented by ADMGs *without* directed mixed cycles. However, it is possible that linear SEMs *with* correlated errors represented by ADMGs *with* directed mixed cycles may imply constraints on the covariance matrix that are not implied by zero partial correlations.

Although the intended application is in linear SEMs, the local Markov properties presented in the paper are valid for ADMGs associated with any probability distributions that satisfy the composition axiom. For example, any probability distribution that is faithful[5] to some DAG or undirected graph (and the marginals of the distribution) satisfies the composition axiom.

Model debugging for ADMGs using vanishing partial correlations is another area of current research. In this model debugging problem, the goal is to modify a graph based

5. A probability distribution P is said to be faithful to a graph G if all the conditional independence relations embedded in P are encoded in G (via the global Markov property).

on the pattern of rejected hypotheses. The properties of ADMGs presented in this paper may facilitate the development of a new model debugging method.

Acknowledgments

We thank the anonymous reviewers for helpful comments. This research was partly supported by NSF grant IIS-0347846.

References

S.A. Andersson, D. Madigan, and M.D. Perlman. Alternative Markov properties for chain graphs. *Scandinavian Journal of Statistics*, 28:33–86, 2001.

K.A. Bollen. *Structural Equations with Latent Variables*. John Wiley, New York, 1989.

D.R. Cox and N. Wermuth. Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218, 1993.

M. Drton and M.D. Perlman. Multiple testing and error control in gaussian graphical model selection. *Statistical Science*, 22(3):430–449, 2007.

O.D. Duncan. *Introduction to Structural Equation Models*. Academic Press, New York, 1975.

M. Frydenberg. The chain graph markov property. *Scandinavian Journal of Statistics*, 17: 333–353, 1990.

A.S. Goldberger. Structural equation models in the social sciences. *Econometrica: Journal of the Econometric Society*, 40:979–1001, 1972.

T. Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11:1–12, 1943. Reprinted in D.F. Hendry and M.S. Morgan (Eds.), *The Foundations of Econometric Analysis*, Cambridge University Press, 477–490, 1995.

G. Kauermann. On a dualization of graphical Gaussian models. *Scandinavian Journal of Statistics*, 23:105–116, 1996.

J.T.A. Koster. On the validity of the Markov interpretation of path diagrams of gaussian structural equations systems with correlated errors. *Scandinavian Journal of Statistics*, 26:413–431, 1999.

S.L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

S.L. Lauritzen and N. Wermuth. Graphical models for association between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.

S.L. Lauritzen, A.P. Dawid, B.N. Larsen, and H.G. Leimer. Independence properties of directed Markov fields. *Networks*, 20:491–505, 1990.

R.P. McDonald. What can we learn from the path equations?: Identifiability, constraints, equivalence. *Psychometrika*, 67(2):225–249, 2002.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

J. Pearl. Graphs, causality, and structural equation models. *Socioligical Methods and Research*, 27:226–284, 1998.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, NY, 2000.

J. Pearl and P. Meshkat. Testing regression models with fewer regressors. In *Proceedings of AI-STAT*, pages 255–259, 1999.

T. Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *Annals of Statistics*, 30(4): 962–1030, 2002.

B. Shipley. A new inferential test for path models based on directed acyclic graphs. *Structural Equation Modeling*, 7:206–218, 2000.

B. Shipley. Testing recursive path models with correlated errors using d-separation. *Structural Equation Modeling*, 10:214–221, 2003.

P. Spirtes, T. Richardson, C. Meek, R. Scheines, and C. Glymour. Using path diagrams as a structural equation modeling tool. *Socioligical Methods and Research*, 27:182–225, 1998.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2001.

N. Wermuth and D.R. Cox. Graphical models: overview. *International Encyclopedia of the Social and Behavioral Sciences*, 9:6379–6386, 2001.

N. Wermuth and D.R. Cox. Joint response graphs and separation induced by triangular systems. *Journal of the Royal Statistical Society B*, 66:687–717, 2004.

S. Wright. The method of path coefficients. *Ann. Math. Statist.*, 5:161–215, 1934.

JMLR: Workshop and Conference Proceedings 10:301–340, 2009 Submitted 4/08; Revised 1/09; Published 2/09

Improving the Reliability of Causal Discovery from Small Data Sets Using Argumentation

Facundo Bromberg BROMBERG@CS.IASTATE.EDU
Dimitris Margaritis DMARG@CS.IASTATE.EDU
Dept. of Computer Science
Iowa State University
Ames, IA 50011

Editor: Constantin Aliferis

Abstract

We address the problem of improving the reliability of independence-based causal discovery algorithms that results from the execution of statistical independence tests on small data sets, which typically have low reliability. We model the problem as a knowledge base containing a set of independence facts that are related through Pearl's well-known axioms. Statistical tests on finite data sets may result in errors in these tests and inconsistencies in the knowledge base. We resolve these inconsistencies through the use of an instance of the class of defeasible logics called argumentation, augmented with a preference function, that is used to reason about and possibly correct errors in these tests. This results in a more robust conditional independence test, called an *argumentative independence test*. Our experimental evaluation shows clear positive improvements in the accuracy of argumentative over purely statistical tests. We also demonstrate significant improvements on the accuracy of causal structure discovery from the outcomes of independence tests both on sampled data from randomly generated causal models and on real-world data sets.

Keywords: independence-based causal discovery, causal Bayesian networks, structure learning, argumentation, reliability improvement

1. Introduction and Motivation

Directed graphical models, also called *Bayesian networks*, can be used to represent the probability distribution of a domain. This makes them a useful and important tool for machine learning where a common task is inference, that is, predicting the probability distribution of a variable of interest given some other knowledge, usually in the form of values of other variables in the domain. An additional use of Bayesian networks comes by augmenting them with causal semantics that represent cause and effect relationships in the domain. The resulting networks are called *causal*. An important problem is inferring the structure of these networks, a process that is sometimes called *causal discovery*, which can provide insights into the underlying data generation process.

Two major classes of algorithms exist for learning the structure of Bayesian networks. One class contains so-called *score-based* methods, which learn the structure by conducting a search in the space of all structures in an attempt to find the structure of maximum score. This score is usually penalized log-likelihood, for example, the Bayesian Information Criterion (BIC) or the (equivalent) Minimum Description Length (MDL). A second class of algorithms works by exploiting the fact that a causal Bayesian

network implies the existence of a set of conditional independence statements between sets of domain variables. Algorithms in this class use the outcomes of a number of conditional independences to constrain the set of possible structures consistent with these to a singleton (if possible) and infer that structure as the only possible one. As such they are called *constraint-based* or *independence-based* algorithms. In this paper we address open problems related to the latter class of algorithms.

It is well-known that independence-based algorithms have several shortcomings. A major one has to do with the effect that unreliable independence information has on the their output. In general such independence information comes from two sources: (a) a domain expert that can provide his or her opinion on the validity of certain conditional independences among some of the variables, sometimes with a degree of confidence attached to them, and/or (b) statistical tests of independence, conducted on data gathered from the domain. As expert information is often costly and difficult to obtain, (b) is the most commonly used option in practice. A problem that occurs frequently however is that the data set available may be small. This may happen for various reasons: lack of subjects to observe (e.g., in medical domains), an expensive data-gathering process, privacy concerns and others. Unfortunately, the reliability of statistical tests significantly diminishes on small data sets. For example, Cochran (1954) recommends that Pearson's χ^2 independence test be deemed unreliable if more than 20% of the cells of the test's contingency table have an expected count of less than 5 data points. Unreliable tests, besides producing errors in the resulting causal model structure, may also produce cascading errors due to the way that independence-based algorithms work: their operation, including which test to evaluate next, typically depends on the outcomes of previous ones. Thus a single error in a statistical test can be propagated by the subsequent choices of tests to be performed by the algorithm, and finally when the edges are oriented. Therefore, an error in a previous test may have large (negative) consequences in the resulting structure, a property that is called *instability* in Spirtes et al. (2000). One possible method for addressing the effect of multiple errors in the construction of a causal model through multiple independence tests is the Bonferroni correction (Hochberg, 1988; Abdi, 2007), which works by dividing the type I error probability α of each test by the number of such tests evaluated during the entire execution of the causal learning algorithm. As a result, the collective type I error probability (of all tests evaluated) is α, that is, 0.05 typically. However, this may make the detection of true dependences harder, as now larger data sets would be required to reach the adjusted confidence threshold of each test. The types of adjustments that may be appropriate for each case to tests that may be dependent is an open problem and the subject of current research in statistics (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001; Storey, 2002).

In this paper we present and evaluate a number of methods for increasing the reliability of independence tests for small data sets. A result of this is the improvement in reliability of independence-based causal discovery algorithms that use these data sets, as we demonstrate in our experiments. We model this setting as a knowledge base whose contents are propositions representing conditional independences that may contain errors. Our main insight is to recognize that the outcomes of independence tests are not themselves independent but are constrained by the outcomes of other tests through Pearl's well-known properties of the conditional independence relation (Pearl, 1988; Dawid, 1979). These can therefore be seen as *integrity constraints* that can correct certain inconsistent test outcomes, choosing instead the outcome that can be inferred by tests that do not result in contradictions. We illustrate this by an example.

Example 1 *Consider an independence-based knowledge base that contains the following propositions, obtained through statistical tests on data.*

$$(\{0\} \perp\!\!\!\perp \{1\} \mid \{2\}) \tag{1}$$
$$(\{0\} \not\perp\!\!\!\perp \{3\} \mid \{2\}) \tag{2}$$
$$(\{0\} \perp\!\!\!\perp \{3\} \mid \{1,2\}) \tag{3}$$

where $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ *denotes conditional independence of the set of variables* $\mathbf{X}$ *with* $\mathbf{Y}$ *conditional on set* $\mathbf{Z}$*, and* $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ *denotes conditional dependence. Suppose that (3) is in fact wrong. Such an error can be avoided if there exists a constraint involving these independence propositions. For example, suppose that we also know that the following rule holds in the domain (this is an instance of an application of the Contraction and Decomposition axioms, described later in Section 2):*

$$(\{0\} \perp\!\!\!\perp \{1\} \mid \{2\}) \wedge (\{0\} \not\perp\!\!\!\perp \{3\} \mid \{2\}) \implies (\{0\} \not\perp\!\!\!\perp \{3\} \mid \{1,2\}). \tag{4}$$

Rule (4), together with independence proposition (1) and dependence proposition (2), contradict independence proposition (3), resulting in an inconsistent knowledge base. If Rule (4) and propositions (1) and (2) are accepted, then proposition (3) must be rejected (and its value reversed), correcting the error in this case. The framework presented in the rest of the paper provides a principled approach for resolving such inconsistencies.

The situation described in the previous example, while simple, illustrates the general idea that we will use in the rest of the paper: the set of independences and dependences used in a causal discovery algorithm form a potentially inconsistent knowledge base, and making use of general rules, derived from axioms and theorems that we know hold in the domain, helps us correct certain outcomes of statistical tests. In this way we will be able to improve the reliability of causal discovery algorithms that use them to derive causal models. To accomplish this we use the framework of *argumentation*, which provides a sound and elegant way of resolving inconsistencies in such knowledge bases, including ones that contain independences.

The rest of the paper is organized as follows. The next section introduces our notation and definitions. Section 3 presents the argumentation framework and its extension with preferences, and describes our approach for applying it to represent and reason in knowledge bases containing independence facts that may be inconsistent. Section 4 introduces the argumentative independence test, implemented by the top-down algorithm introduced in Section 5. We then present an approximation for the top-down algorithm in Section 6 that reduces its time complexity to polynomial. We experimentally evaluate our approach in Section 7, and conclude with a summary and possible directions of future research in Section 8. Most of the proofs are presented in detail in Appendices A and B, which contain proofs for the computability (termination) and the validity (no AIT test can return a dependence and an independence result at the same time) of AIT, respectively. Note that, as our main goal in this paper is to address the problem of robust causal learning and not necessarily to advance the theory of argumentation itself, our exposition in the rest of the paper is geared toward causality theorists and practitioners. As this community may be unfamiliar with the theory and methods of the argumentation framework, we have included a self-contained discussion that covers the basic definitions and theorems of argumentation theory in some detail.

2. Notation and Preliminaries

In this work we denote random variables with capitals (e.g., X, Y, Z) and sets of variables with bold capitals (e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$). In particular, we denote by $\mathbf{V} = \{1, \ldots, n\}$ the set of all n variables in the domain, naming the variables by their indices in $\mathbf{V}$; for instance, we refer to the third variable in $\mathbf{V}$ simply by 3. We assume that all variables in the domain are discrete following a multinomial distribution or are continuous following a Gaussian distribution. We denote the data set by D and its size (number of data points) by N. We use the notation $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ to denote that the variables in set $\mathbf{X}$ are (jointly) independent of those in $\mathbf{Y}$ conditional on the values of the variables in $\mathbf{Z}$, for disjoint sets of variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, while $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ denotes conditional dependence. For the sake of readability, we slightly abuse this notation and use $(X \perp\!\!\!\perp Y \mid Z)$ as shorthand for $(\{X\} \perp\!\!\!\perp \{Y\} \mid \{Z\})$.

A Bayesian network (**BN**) is a directed graphical model which represents the joint probability distribution over $\mathbf{V}$. Each node in the graph represents one of the random variables in the domain. The structure of the network implicitly represents a set of conditional independences on the domain variables. Given the structure of a BN, the set of independences implied by it can be identified by a process called *d-separation* (Pearl, 1988); the latter follows from the *local Markov property* that states that each node in the network is conditionally independent of all its non-descendants in the graph given its parents. All independences identified by d-separation are implied by the model structure. If, in addition, all remaining triplets $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ correspond to dependencies, we say that the BN is *directed graph-isomorph* (abbreviated *DAG-isomorph*) or simply *causal* (as defined by Pearl, 1988). The concept of DAG-isomorphism is equivalent to a property called Faithfulness in Spirtes et al. (2000). A graph G is said to be *faithful* to some distribution if exactly those independences that exist in the distribution and no others are returned by the process of d-separation on G. In this paper we assume Faithfulness. For learning the structure of the Bayesian network of a domain we make use of the PC algorithm (Spirtes et al., 2000), which is only able to correctly identify the structure under the assumption of *causal sufficiency*. We therefore also assume causal sufficiency. A domain is causally sufficient if it does not contain any *hidden* or *latent* variables.

As mentioned above, independence-based algorithms operate by conducting a series of conditional independence queries. For these we assume that an *independence-query oracle* exists that is able to provide such information. This approach can be viewed as an instance of the statistical query oracle theory of Kearns and Vazirani (1994). In practice such an oracle does not exist, but can be implemented approximately by a statistical test evaluated on the data set (for example, this can be Pearson's conditional independence χ^2 (chi-square) test (Agresti, 2002), Wilk's G^2 test, a mutual information test etc.). In this work we used Wilk's G^2 test (Agresti, 2002). To determine conditional independence between two variables X and Y given a set $\mathbf{Z}$ from data, the statistical test G^2 (and many other independence tests based on hypothesis testing, for example, the χ^2 test) uses the values in the contingency table (a table containing the data point counts for each possible combination of the variables that participate in the test) to compute a *test statistic*. For a given value of the test statistic, the test then computes the likelihood of obtaining that or a more extreme value by chance under the *null hypothesis*, which in our case is that the two variables are conditionally independent. This likelihood, called the *p-value* of the test, is then returned. The p-value of a test equals the probability of falsely rejecting the null hypothesis (independence). Assuming that the p-value of a test

$$\begin{array}{rl}
\textbf{(Symmetry)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \iff (\mathbf{Y} \perp\!\!\!\perp \mathbf{X} \mid \mathbf{Z}) \\
\textbf{(Decomposition)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Weak Union)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \\
\textbf{(Contraction)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Intersection)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z})
\end{array} \tag{5}$$

$$\begin{array}{rl}
\textbf{(Symmetry)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \iff (\mathbf{Y} \perp\!\!\!\perp \mathbf{X} \mid \mathbf{Z}) \\
\textbf{(Composition)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Decomposition)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Intersection)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Weak Union)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \\
\textbf{(Contraction)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}) \\
\textbf{(Weak Transitivity)} & (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma) \implies (\mathbf{X} \perp\!\!\!\perp \gamma \mid \mathbf{Z}) \vee (\gamma \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \\
\textbf{(Chordality)} & (\alpha \perp\!\!\!\perp \beta \mid \gamma \cup \delta) \wedge (\gamma \perp\!\!\!\perp \delta \mid \alpha \cup \beta) \implies (\alpha \perp\!\!\!\perp \beta \mid \gamma) \vee (\alpha \perp\!\!\!\perp \beta \mid \delta)
\end{array} \tag{6}$$

is $p(X, Y \mid \mathbf{Z})$, the statistical test concludes independence if and only if $p(X, Y \mid \mathbf{Z})$ is greater than a threshold α, that is,

$$(X \perp\!\!\!\perp Y \mid \mathbf{Z}) \iff p(X, Y \mid \mathbf{Z}) > \alpha.$$

Common values in statistics for α are 0.05 and 0.01, corresponding to *confidence thresholds* $(1 - \alpha)$ of 0.95 and 0.99 respectively. The value 0.10 for α is also sometimes used, depending on the application, while values as low as 0.005 and 0.001 are sometimes used for structure learning.

The conditional independences and dependences of a domain are connected through a set of general rules, introduced in Pearl (1988) and shown boxed in Eq. (5). These can be seen as constraints in a meta-space representing all possible independences in the domain. More specifically, let us imagine a meta-space of binary variables, each corresponding to the truth value of the independence of a triplet $(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$ (e.g., `true` for independence and `false` for dependence). Each point in this space corresponds to a conditional independence assignment to all possible triplets in the domain. In this conceptual space not all points are tenable; in particular the set of rules of Eq. (5) constrain the truth values of independences corresponding to triplets. For domains for which there exists a faithful Bayesian network a more relaxed set of properties hold, shown boxed in Eq. (6) where α, β, γ and δ correspond to single variables. In both sets of axioms, the property of Intersection holds if the probability distribution of the domain is positive, meaning that every assignment to all variables in the domain has a non-zero probability. Eq. (6) were first introduced by Dawid (1979) in a slightly different form and independently re-discovered by Pearl and Paz (1985).

Note that the axioms of Eq. (5) are necessarily incomplete; Studený (1991) showed that there is no finite axiomatization of the conditional independence relation in general. The implication of this is that there may be some inconsistencies involving some set of independences and dependences that no method can detect and resolve.

In the next section we describe the argumentation framework, which allows one to make beneficial use of these constraints. This is followed by its application to our problem of answering independence queries from knowledge bases that contain sets of potentially inconsistent independence propositions.

3. The Argumentation Framework

There exist two major approaches for reasoning with information contained in inconsistent knowledge bases such as those containing independence statements that were described in the previous section. These two distinct approaches correspond to two different attitudes: One is to resolve the inconsistencies by removing a subset of propositions such that the resulting KB becomes consistent; this is called *belief revision* in the literature (Gärdenfors, 1992; Gärdenfors and Rott, 1995; Shapiro, 1998; Martins, 1992). A potential shortcoming (Shapiro, 1998) of belief revision stems from the fact that it removes propositions, which discards potentially valuable information. In addition, an erroneous modification of the KB (such as the removal of a proposition) may have unintended negative consequences if later more propositions are inserted in the KB. A second approach to inconsistent KBs is to allow inconsistencies but to use rules that may be possibly contained in it to deduce which truth value of a proposition query is "preferred" in some way. One instance of this approach is *argumentation* (Dung, 1995; Loui, 1987; Prakken, 1997; Prakken and Vreeswijk, 2002), which is a sound approach that allows inconsistencies but uses a proof procedure that is able to deduce (if possible) that one of the truth values of certain propositions is preferred over its negation. Argumentation is a reasoning model that belongs to the broader class of defeasible logics (Pollock, 1992; Prakken, 1997). Our approach uses the argumentation framework of Amgoud and Cayrol (2002) that considers preferences over arguments, extending Dung's more fundamental framework (Dung, 1995). Preference relations give an extra level of specificity for comparing arguments, allowing a more refined form of selection between conflicting propositions. Preference-based argumentation is presented in more detail in Section 3.2.

We proceed now to describe the argumentation framework.

Definition 1 *An* argumentation framework *is a pair* $\langle \mathcal{A}, \mathcal{R} \rangle$*, where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R}$ *is a binary relation representing a defeasibility relationship between arguments, that is,* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$*.* $(a, b) \in \mathcal{R}$ *or equivalently "a* $\mathcal{R}$ *b" reads that argument a defeats argument b. We also say that a and b are in conflict.*

An example of the defeat relation $\mathcal{R}$ is *logical defeat*, which occurs when an argument contradicts another logically.

The elements of the argumentation framework are not propositions but *arguments*. Given a potentially inconsistent knowledge base $\mathcal{K} = \langle \Sigma, \Psi \rangle$ with a set of propositions Σ and a set of inference rules Ψ, arguments are defined formally as follows.

Definition 2 *An* argument *over knowledge base* $\langle \Sigma, \Psi \rangle$ *is a pair* (H, h) *where* $H \subseteq \Sigma$ *such that:*

- H *is consistent,*
- $H \vdash_\Psi h$*,*
- H *is minimal (with respect to set inclusion).*

H *is called the* support *and* h *the* conclusion *or* head *of the argument.*

In the above definition $\vdash_\Psi$ stands for classical logical inference over the set of inference rules Ψ. Intuitively an argument (H, h) can be thought as an "if-then" rule, that is, "if H then h." In inconsistent knowledge bases two arguments may contradict or *defeat* each other. The defeat relation is defined through the *rebut* and *undercut* relations, defined as follows.

Algorithm 1: Recursive computation of acceptable arguments: $Acc_{\mathcal{R}} = \mathcal{F}(\mathcal{A}, \mathcal{R}, S)$

1: $S' \longleftarrow S \cup \{a \in \mathcal{A} \mid a \text{ is defended by } S\}$
2: **if** $S = S'$ **then**
3: **return** S'
4: **else**
5: **return** $\mathcal{F}(\mathcal{A}, \mathcal{R}, S')$
6: **end if**

Definition 3 *Let* (H_1, h_1), (H_2, h_2) *be two arguments.*

- (H_1, h_1) rebuts (H_2, h_2) *iff* $h_1 \equiv \neg h_2$.
- (H_1, h_1) undercuts (H_2, h_2) *iff* $\exists h \in H_2$ *such that* $h \equiv \neg h_1$.

If (H_1, h_1) *rebuts or undercuts* (H_2, h_2) *we say that* (H_1, h_1) defeats (H_2, h_2).

(The symbol "$\equiv$" stands for logical equivalence.) In other words, $(H_1, h_1)\ \mathcal{R}\ (H_2, h_2)$ if and only if (H_1, h_1) *rebuts* or *undercuts* (H_2, h_2).

The objective of argumentation is to decide on the acceptability of a given argument. There are three possibilities: an argument can be accepted, rejected, or neither. This partitions the space of arguments $\mathcal{A}$ in three classes:

- The class $Acc_{\mathcal{R}}$ of *acceptable arguments*. Intuitively, these are the "good" arguments. In the case of an inconsistent knowledge base, these will be inferred from the knowledge base.
- The class $Rej_{\mathcal{R}}$ of *rejected arguments*. These are the arguments defeated by acceptable arguments. When applied to an inconsistent knowledge base, these will not be inferred from it.
- The class $Ab_{\mathcal{R}}$ of arguments *in abeyance*. These arguments are neither accepted nor rejected.

The semantics of acceptability proposed by Dung (1995) dictates that an argument should be accepted if it is not defeated, or if it is defended by acceptable arguments, that is, each of its defeaters is itself defeated by an acceptable argument. This is formalized in the following definitions.

Definition 4 *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an argumentation framework, and* $S \subseteq \mathcal{A}$. *An argument* a *is* defended *by* S *if and only if* $\forall b$, *if* $(b\ \mathcal{R}\ a)$ *then* $\exists c \in S$ *such that* $(c\ \mathcal{R}\ b)$.

Dung characterizes the set of acceptable arguments by a monotonic function $\mathcal{F}$, that is, $\mathcal{F}(S) \subseteq \mathcal{F}(S \cup T)$ for some S and T. Given a set of arguments $S \subseteq \mathcal{A}$ as input, $\mathcal{F}$ returns the set of all arguments defended by S:

Definition 5 *Let* $S \subseteq \mathcal{A}$. *Then* $\mathcal{F}(S) = \{a \in \mathcal{A} \mid a \text{ is defended by } S\}$.

Slightly overloading our notation, we define $\mathcal{F}(\varnothing)$ to contain the set of arguments that are not defeated by any argument in the framework.

Definition 6 $\mathcal{F}(\varnothing) = \{a \in \mathcal{A} \mid a \text{ is not defeated by any argument in } \mathcal{A}\}$.

Dung proved that the set of acceptable arguments is the least fix-point of $\mathcal{F}$, that is, the smallest set S such that $\mathcal{F}(S) = S$.

Theorem 1 (Dung 1995) *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework. The set of acceptable arguments $Acc_{\mathcal{R}}$ is the least fix-point of the function $\mathcal{F}$.*

Dung also showed that if the argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ is finitary, that is, for each argument A there are finitely many arguments that defeat A, the least fix-point of function $\mathcal{F}$ can be obtained by iterative application of $\mathcal{F}$ to the empty set. We can understand this intuitively: From our semantics of acceptability it follows that all arguments in $\mathcal{F}(\varnothing)$ are accepted. Also, every argument in $\mathcal{F}(\mathcal{F}(\varnothing))$ must be acceptable as well since each of its arguments is defended by acceptable arguments. This reasoning can be applied recursively until a fix-point is reached. This happens when the arguments in S cannot be used to defend any other argument not in S, that is, no additional argument is accepted. This suggests a simple algorithm for computing the set of acceptable arguments. Algorithm 1 shows a recursive procedure for this, based on the above definition. The algorithm takes as input an argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ and the set S of arguments found acceptable so far, that is, $S = \varnothing$ initially.

Let us illustrate these ideas with an example.

Example 2 *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework defined by $\mathcal{A} = \{a, b, c\}$ and $\mathcal{R} = \{(a, b),$*
$(b, c)\}$. The only argument that is not defeated is a, and therefore $\mathcal{F}(\varnothing) = \{a\}$. Argument b is defeated by the acceptable argument a, so b cannot be defended and is therefore rejected, that is, $b \in Rej_{\mathcal{R}}$. Argument c, though defeated by b, is defended by (acceptable argument) a which defeats b, so c is acceptable. The set of acceptable arguments is therefore $Acc_{\mathcal{R}} = \{a, c\}$ and the set of rejected arguments is $Rej_{\mathcal{R}} = \{b\}$.

The **bottom-up** approach of Algorithm 1 has the disadvantage that it requires the computation of all acceptable arguments to answer the acceptability status of a single one. In practice, and in particular in the application of argumentation to independence tests, the entire set of acceptable arguments is rarely needed. An alternative is to take a top-down approach (Amgoud and Cayrol, 2002; Dung, 1995; Toni and Kakas, 1995; Kakas and Toni, 1999) that evaluate the acceptability of some input argument by evaluating (recursively) the acceptability of its attackers. Below we present an alternative algorithm, called the **top-down algorithm**, for deciding the acceptability of an input argument. This algorithm is a version of the *dialog tree* algorithm of Amgoud and Cayrol (2002), where details unnecessary for the current exposition are not shown. This algorithm is provably equivalent to Algorithm 1 (whenever it is given the same input it is guaranteed to produce the same output), but it is considerably more efficient (as shown later in Section 5.2). We sketch the algorithm here and show a concrete version using the preference-based argumentation framework in Section 3.2.

Given an input argument a, the top-down algorithm employs a goal-driven approach for answering whether a is accepted or not. Its operation is guided by the

Algorithm 2: Top-down computation of acceptable arguments: *top-down*$(\mathcal{A}, \mathcal{R}, a)$

```
1: defeaters ← set of arguments in A that defeat a according to R.
2: for d ∈ defeaters do
3:    if top-down(A, R, d) = accepted then
4:       return rejected
5:    end if
6: end for
7: return accepted
```

same acceptability semantics as those used for Algorithm 1. Let us denote the predicates $A(a) \equiv (a \in Acc_{\mathcal{R}})$, $R(a) \equiv (a \in Rej_{\mathcal{R}})$, and $Ab(a) \equiv (a \in Ab_{\mathcal{R}})$. Then, the acceptability semantics are as follows.

(Acceptance) A node is accepted iff it has no defeaters or all its defeaters are rejected:

$$A(a) \iff \forall b \in defeaters(a), R(b).$$

(Rejection) A node is rejected iff at least one of its defeaters is accepted:

$$R(a) \iff \exists b \in defeaters(a), A(b). \tag{7}$$

(Abeyance) A node is in abeyance iff its not accepted nor rejected:

$$Ab(a) \iff \neg A(a) \wedge \neg R(a).$$

The logic of these equations can be easily implemented with a recursive algorithm, shown in Algorithm 2. The algorithm, given some input argument a, loops over all defeaters of a and responds `rejected` if any of its defeaters is accepted (line 4). If execution reaches the end of the loop at line 7 then that means that none of its defeaters was accepted, and thus the algorithm accepts the input argument a. We can represent the execution of the top-down algorithm graphically by a tree that contains a at the root node, and all the defeaters of a node as its children. A leaf is reached when a node has no defeaters. In that case the loop contains no iterations and line 7 is reached trivially.

Unfortunately, the top-down algorithm, as shown in Algorithm 2, will fail to terminate when a node is in abeyance. This is clear from the following lemma (proved formally in Appendix A but reproduced here to aid our intuition).

Lemma 7 *For every argument a,*

$$Ab(a) \implies \exists b \in attackers(a), Ab(b).$$

(An attacker is a type of defeater; it is explained in detail in the next section. For the following discussion the reader can substitute "attacker" with "defeater" in the lemma above.) From this lemma we can see that, if an argument is in abeyance, its set of defeaters must contain an argument in abeyance and thus the recursive call of the top-down algorithm will never terminate, as there will always be another defeater in abeyance during each call. While there are ways to overcome this difficulty in the general case, we can prove that using the preference-based argumentation framework (presented later in the paper) and for the particular preference relation introduced for deciding on independence tests (c.f. Section 3.3), no argument can be in abeyance and thus the top-down algorithm always terminates. A formal proof of this is presented later in Section 5.

We conclude the section by proving that the top-down algorithm is equivalent to the bottom-up algorithm of Algorithm 1 that is, given the same input as Algorithm 1 it is guaranteed to produce the same output. The proof assumes no argument is in abeyance. This assumption is satisfied for argumentation in independence knowledge bases (c.f. Theorem 5, Section 5).

Theorem 2 *Let a be an argument in the argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$, and let $\mathcal{F}$ be the set of acceptable arguments output by Algorithm 1. Assuming a is not in abeyance,*

$$\textit{top-down}(\mathcal{A}, \mathcal{R}, a) = \texttt{accepted} \iff a \in \mathcal{F} \tag{8}$$

$$\textit{top-down}(\mathcal{A}, \mathcal{R}, a) = \texttt{rejected} \iff a \notin \mathcal{F}. \tag{9}$$

Proof According to Theorem 1, the fix point of function $\mathcal{F}$ returned by Algorithm 1 contains the set of arguments considered acceptable by the acceptability semantics of Dung. As the top-down algorithm is a straightforward implementation of Dung's acceptability semantics expressed by Eq. (7), the double implication of Eq. (8) must follow. To prove Eq. (9) we can prove the equivalent expression with both sides negated, that is,

$$\textit{top-down}(\mathcal{A}, \mathcal{R}, a) \neq \texttt{rejected} \iff a \in \mathcal{F}.$$

Since a is not in abeyance, if the top-down algorithm does not return `rejected` it must return `accepted`. The double implication is thus equivalent to Eq. (8), which was proved true. ■

3.1. Argumentation in Independence Knowledge Bases

We can now apply the argumentation framework to our problem of answering queries from knowledge bases that contain a number of potentially inconsistent independences and dependencies and a set of rules that express relations among them.

Definition 8 *An* independence knowledge base *(**IKB**) is a knowledge base $\langle \Sigma, \Psi \rangle$ such that its set of propositions Σ contains independence propositions of the form $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ or $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ for $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ disjoint subsets of $\mathbf{V}$, and its set of inference rules Ψ is either the general set of axioms shown in Eq. (5) or the specific set of axioms shown in Eq. (6).*

For IKBs, the set of arguments $\mathcal{A}$ is obtained in two steps. First, for each proposition $\sigma \in \Sigma$ (independence or dependence) we add to $\mathcal{A}$ the argument $(\{\sigma\}, \sigma)$. This is a valid argument according to Definition 2 since its support $\{\sigma\}$ is (trivially) consistent, it (trivially) implies the head σ, and it is minimal (the pair $(\varnothing, \sigma)$ is not a valid argument since $\varnothing$ is equivalent to the proposition `true` which does not entail σ in general). We call arguments of the form $(\{\sigma\}, \sigma)$ *propositional arguments* since they correspond to single propositions. The second step in the construction of the set of arguments $\mathcal{A}$ concerns rules. Based on the chosen set of axioms (general or directed) we construct an alternative, logically equivalent set of rules Ψ', each member of which is *single-headed*, that is, contains a single proposition as the consequent, and *decomposed*, that is, each of its propositions is an independence statement over single variables (the last step is justified by the fact that typical algorithms for causal learning never produce nor require the evaluation of independence between sets).

To construct the set of single-headed rules we consider, for each axiom, all possible contrapositive versions of it that have a single head. To illustrate, consider the Weak Transitivity axiom

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma) \implies (\mathbf{X} \perp\!\!\!\perp \gamma \mid \mathbf{Z}) \vee (\gamma \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$$

from which we obtain the following set of single-headed rules:

$$\begin{aligned}
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma) \wedge (\mathbf{X} \not\perp\!\!\!\perp \gamma \mid \mathbf{Z}) &\implies (\gamma \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma) \wedge (\gamma \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) &\implies (\mathbf{X} \perp\!\!\!\perp \gamma \mid \mathbf{Z}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma) \wedge (\gamma \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \not\perp\!\!\!\perp \gamma \mid \mathbf{Z}) &\implies (\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\gamma \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \not\perp\!\!\!\perp \gamma \mid \mathbf{Z}) &\implies (\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \gamma).
\end{aligned}$$

To obtain decomposed rules we apply the Decomposition axiom to every single-headed rule to produce only propositions over singletons. To illustrate, consider the Intersection axiom:

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z}).$$

In the above the consequent coincides with the antecedent of the Decomposition axiom, and we thus replace the Intersection axiom with a decomposed version:

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z}).$$

Finally, note that it is easy to show that this rule is equivalent to two single-headed rules, one implying $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ and the other implying $(\mathbf{X} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z})$.

The result of the application of the above procedures is a set of single-headed, decomposed rules Ψ'. We construct, for each such rule $(\Phi_1 \wedge \Phi_2 \ldots \wedge \Phi_n \implies \varphi) \in \Psi'$ and for each subset of Σ that matches exactly the set of antecedents, that is, each subset $\{\varphi_1, \varphi_2 \ldots, \varphi_n\}$ of Σ such that $\Phi_1 \equiv \varphi_1, \Phi_2 \equiv \varphi_2 \ldots \Phi_n \equiv \varphi_n$, the argument $(\{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n\}, \varphi)$, and add it to $\mathcal{A}$.[1]

IKBs can be augmented with a set of preferences that allow one to take into account the reliability of each test when deciding on the truth value of independence queries. This is described in the next section.

3.2. Preference-based Argumentation Framework

Following Amgoud and Cayrol (2002), we now refine the argumentation framework of Dung (1995) for cases where it is possible to define a preference order Π over arguments.

Definition 9 *A* preference-based argumentation framework *(**PAF**) is a triplet* $\langle \mathcal{A}, \mathcal{R}, \Pi \rangle$ *where* $\mathcal{A}$ *is a set of arguments,* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary relation representing a defeat relationship between pairs of arguments, and* Π *is a (partial or total) order over* $\mathcal{A}$.

For the case of inconsistent knowledge bases, preference Π over arguments follows the preference π over their support, that is, stronger support implies a stronger argument, which is given as a partial or total order over sets of propositions. Formally:

Definition 10 *Let* $\mathcal{K} = \langle \Sigma, \Psi \rangle$ *be a knowledge base,* π *be a (partial or total) order on subsets of* Σ *and* (H, h), (H', h') *two arguments over* $\mathcal{K}$. *Argument* (H, h) *is* π-preferred *to* (H', h') *(denoted* $(H, h) \gg_\pi (H', h')$*) if and only if* H *is preferred to* H' *with respect to* π.

In what follows we overload our notation by using π to denote either the ordering over arguments or over their supports.

An important sub-class of preference relations is the *strict* and *transitive* preference relation, defined as follows.

Definition 11 *We say that preference relation* π *over arguments is* strict *if the order of arguments induced by it is strict and total, that is, for every pair of arguments* a *and* b,

$$(a \gg_\pi b) \iff \neg(b \gg_\pi a).$$

1. This is equivalent to propositionalizing the set of rules, which are first-order (the rules of Eqs. (5) and (6) are universally quantified over all sets of variables, and thus are the rules in Ψ'). As this may be expensive (exponential in the number of propositions), in practice it is not implemented in this way; instead, appropriate rules are matched on the fly during the argumentation inference process.

Definition 12 *We say that preference relation π over arguments is* transitive *if, for every three arguments a, b and c,*

$$\big(a \gg_\pi b\big) \wedge \big(b \gg_\pi c\big) \implies \big(a \gg_\pi c\big).$$

The importance of the properties of strictness and transitivity will become clear later when we talk about the correctness of the argumentative independence test (defined later in Section 4).

We now introduce the concept of *attack* relation, a combination of the concepts of defeat and preference relation.

Definition 13 *Let $\langle \mathcal{A}, \mathcal{R}, \beta \rangle$ be a PAF, and $a, b \in \mathcal{A}$ be two arguments. We say b* attacks *a if and only if $b \mathcal{R} a$ and $\neg(a \gg_\pi b)$.*

We can see that the attack relation is a special case of the defeat relation and therefore the same conclusions apply; in particular Theorem 1, which allows us to compute the set of acceptable arguments of a PAF using Algorithm 1 or Algorithm 2.

In Sections 3.3 and 4 below, we apply these ideas to construct an approximation to the independence-query oracle that is more reliable than a statistical independence test.

3.3. Preference-based Argumentation in Independence Knowledge Bases

We now describe how to apply the preference-based argumentation framework of Section 3.2 to improve the reliability of conditional independence tests conducted on a (possibly small) data set. A preference-based argumentation framework has three components. The first two, namely $\mathcal{A}$ and $\mathcal{R}$, are identical to the general argumentation framework. We now describe how to construct the third component, namely the preference order π over subsets H of Σ, in IKBs. We define it using a belief estimate $\nu(H)$ that all propositions in H are correct,

$$H \gg_\pi H' \iff \nu(H) > \nu(H') \vee \big[\nu(H) = \nu(H') \wedge f(H, H')\big]. \tag{10}$$

That is, H is preferred over H' if and only if its belief of correctness is higher than that of H' or, in the case that these beliefs are equal, we break the tie using predicate f. For that we require that

$$\forall H, H' \subseteq \mathcal{A}, \text{ such that } H \neq H', \;\; f(H, H') = \neg f(H', H). \tag{11}$$

In addition, we require that f be transitive, that is, $f(H, H') \wedge f(H', H'') \implies f(H, H'')$. This implies that the preference relation π is transitive, which is a necessary condition for proving a number of important theorems in Appendix A. In our implementation we resolved ties by assuming an arbitrary order of the variables in the domain, determined at the beginning of the algorithm and maintained fixed during its entire execution. Based on this ordering, $f(H, H')$ resolved ties by the lexicographic order of the variables in H and H'. By this definition, our f is both non-commutative and transitive.

Before we define $\nu(H)$ we first show that π, as defined by Eqs. (10) and (11) and for any definition of $\nu(H)$, satisfies two important properties, namely strictness (Definition 11) and transitivity (Definition 12). We do this in the following two lemmas.

Lemma 14 *The preference relation for independence knowledge bases defined by Equations (10) and (11) is strict.*

Proof

$$
\begin{aligned}
& H \gg_\pi H' \\
\iff\ & \nu(H) > \nu(H') \vee \big[\nu(H) = \nu(H') \wedge f(H, H')\big] && \textbf{[By Eq. (10)]} \\
\iff\ & \nu(H) \geq \nu(H') \wedge \big[\nu(H) > \nu(H') \vee f(H, H')\big] && \textbf{[Distributivity of } \vee \textbf{ over } \wedge\textbf{]} \\
\iff\ & \neg\big\{\nu(H') > \nu(H) \vee \big[\nu(H') \geq \nu(H) \wedge f(H', H)\big]\big\} && \textbf{[Double negation and Eq. (11)]} \\
\iff\ & \neg\big\{\big[\nu(H') > \nu(H) \vee \nu(H') \geq \nu(H)\big] \wedge \big[\nu(H') > \nu(H) \vee f(H', H)\big]\big\} \\
\iff\ & \neg\big\{\nu(H') \geq \nu(H) \wedge \big[\nu(H') > \nu(H) \vee f(H', H)\big]\big\} \\
\iff\ & \neg\big\{\big[\nu(H') > \nu(H) \vee \nu(H') = \nu(H)\big] \wedge \big[\nu(H') > \nu(H) \vee f(H', H)\big]\big\} \\
\iff\ & \neg\big\{\nu(H') > \nu(H) \vee \big[\nu(H') = \nu(H) \wedge f(H', H)\big]\big\} && \textbf{[Common factor } \nu(H') > \nu(H)\textbf{]} \\
\iff\ & \neg(H' \gg_\pi H) && \textbf{[Again by Eq. (10)]}
\end{aligned}
$$

■

Lemma 15 *The preference relation defined by Equations (10) and (11) is transitive.*

Proof

$$
\begin{aligned}
& H \gg_\pi J \ \wedge\ J \gg_\pi K \\
\iff\ & \Big\{\nu(H) > \nu(J) \ \vee\ \big[\nu(H) = \nu(J) \ \wedge\ f(H, J)\big]\Big\} \\
& \wedge \Big\{\nu(J) > \nu(K) \ \vee\ \big[\nu(J) = \nu(K) \ \wedge\ f(J, K)\big]\Big\} && \textbf{[By Eq. (10)]} \\
\iff\ & \big[\nu(H) > \nu(J) \ \wedge\ \nu(J) > \nu(K)\big] && \textbf{[Case A]} \\
& \vee \big[\nu(H) > \nu(J) \ \wedge\ \nu(J) = \nu(K) \ \wedge\ f(J, K)\big] && \textbf{[Case B]} \\
& \vee \big[\nu(H) = \nu(J) \ \wedge\ f(H, J) \ \wedge\ \nu(J) > \nu(K)\big] && \textbf{[Case C]} \\
& \vee \big[\nu(H) = \nu(J) \ \wedge\ f(H, J) \ \wedge\ \nu(J) = \nu(K) \ \wedge\ f(J, K)\big] && \textbf{[Case D]}
\end{aligned}
$$

To complete the proof we show that each of the cases A, B, C and D implies $H \gg_\pi K$.

(Case A) $\nu(H) > \nu(J) \ \wedge\ \nu(J) > \nu(K) \Longrightarrow \nu(H) > \nu(K) \Longrightarrow H \gg_\pi K$.

(Case B) $\nu(H) > \nu(J) \ \wedge\ \nu(J) = \nu(K) \ \wedge\ f(J, K) \Longrightarrow \nu(H) > \nu(K) \Longrightarrow H \gg_\pi K$.

(Case C) $\nu(H) = \nu(J) \ \wedge\ f(H, J) \ \wedge\ \nu(J) > \nu(K) \Longrightarrow \nu(H) > \nu(K) \Longrightarrow H \gg_\pi K$.

(Case D)

$$\nu(H) = \nu(J) \ \wedge\ f(H, J) \ \wedge\ \nu(J) = \nu(K) \ \wedge\ f(J, K) \Longrightarrow \nu(H) = \nu(K) \ \wedge\ f(H, K) \Longrightarrow H \gg_\pi K,$$

due to the transitivity of predicate f. ■

We now return to the computation of $\nu(H)$. We estimate the belief $\nu(H)$ that a set of propositions H is correct by assuming independence among these propositions.[2] Overloading notation and denoting by $\nu(h)$ the probability of an individual proposition h being correct, the probability of all elements in H being correct under this assumption of independence is

$$\nu(H) = \prod_{h \in H} \nu(h). \tag{12}$$

The belief that a proposition stating independence is correct can be computed in different ways, depending on the particular choice of independence oracle chosen. In this paper we use Wilk's G^2 test, but the resulting belief can be easily adapted to any other independence oracle that produces p-values. We hope that the following discussion serves as a starting point for others to adapt it to other types of independence oracles.

As discussed in Section 2, the p-value $p(X, Y \mid \mathbf{Z})$ computed by this test is the probability of error in rejecting the null hypothesis (conditional independence in our case) and assuming that X and Y are dependent. Therefore, the probability of a test returning dependence of being correct is

$$\nu_D(X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) = 1 - p(X, Y \mid \mathbf{Z})$$

where the subscript D indicates that this expression is valid only for dependencies. Formally, the error of falsely rejecting the null hypothesis is called a *type I error*. To determine the preference of a test returning independence we can, in principle, use this procedure symmetrically: use the probability of error in falsely accepting the null hypothesis (again, this is conditional independence), called a *type II error*, which we denote by $\beta(X, Y \mid \mathbf{Z})$. In this case we can define the preference of independence $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ as the probability of correctly assuming independence by

$$\nu_I(X \perp\!\!\!\perp Y \mid \mathbf{Z}) = 1 - \beta(X, Y \mid \mathbf{Z})$$

where again the subscript I indicates that it is valid only for independences. Unfortunately value of β cannot be obtained without assumptions, because it requires the computation of the probability of the test statistic under the hypothesis of dependence, and there are typically an infinite number of dependent models. In statistical applications, the β value is commonly approximated by assuming one particular dependence model if prior knowledge about that is available. In the absence of such information however in this paper we estimate it using a heuristic function of the p-value, assuming the following heuristic constraints on β:

$$\beta(X, Y \mid \mathbf{Z}) = \begin{cases} 1 & \text{if } p(X, Y \mid \mathbf{Z}) = 0 \\ \alpha - \frac{\alpha}{2+|\mathbf{Z}|} & \text{if } p(X, Y \mid \mathbf{Z}) = 1 \\ \alpha & \text{if } p(X, Y \mid \mathbf{Z}) = \alpha. \end{cases}$$

The first constraint (for $p(X, Y \mid \mathbf{Z}) = 0$) corresponds to the intuition that when the p-value of the test is close to 0, the test statistic is very far from its value under the model that assumes independence, and thus we would give more preference to the

2. The assumption of independence is a heuristic, and is made mainly due to the difficulty of determining the dependence between two or more statistical tests evaluated on the same data set. Other possible ways of defining the preference of a set of propositions are possible. The problem of dealing with multiple tests is an open problem and an area of active research in statistics; see Section 1 for a discussion.

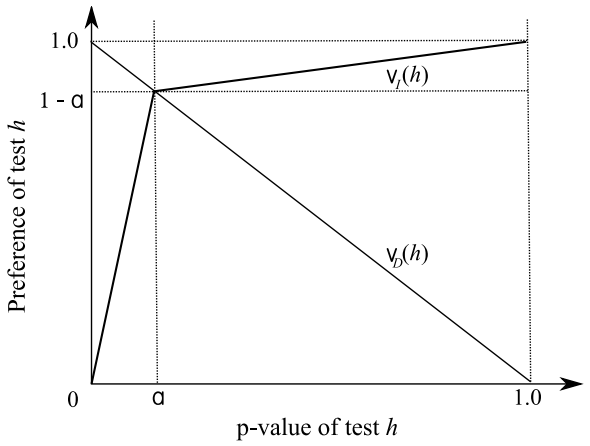


Figure 1: Preference functions $\nu_I(h)$ and $\nu_D(h)$ for statements of independence and dependence respectively, as functions of the p-value of test h.

"dependence" decision. The intuition for the second case ($p(X, Y \mid \mathbf{Z}) = 1$) is reversed—when the value of the statistic is very close to the expected one under independence then independence is preferred. The value of the second case is tempered by the number of variables in the conditioning set. This reflects the practical consideration that, as the number $2 + |\mathbf{Z}|$ of variables involved in the test increases, given a fixed data set, the discriminatory power of the test diminishes as $|\mathbf{Z}| \to \infty$. The third case causes the two functions ν_I and ν_D to intersect at p-value α. This is due to fairness: in the absence of non-propositional arguments (i.e., in the absence of inference rules in our knowledge base), the independence decisions of the argumentation framework should match those of the purely statistical tests, that is, "dependence" if and only if (p-value $\leq \alpha$). If instead we chose a different intersection point, then the resulting change in the outcome of tests may have been simply due to bias in the independence decision that favors dependence or independence, that is, equivalent to an arbitrary change of the threshold of the statistical test, and the comparison of the statistical and the new test based on argumentation would not be a fair one. The remaining values of β are approximated by linear interpolation among the above points. The result is summarized in Fig. 1, which depicts preference functions ν_D and ν_I with respect to the p-value of the corresponding statistical test.

Let us illustrate how the preference-based argumentation can be used to resolve the inconsistencies of Example 1.

Example 3 *In example 1 we considered an IKB with the following propositions*

$$(0 \perp\!\!\!\perp 1 \mid 2) \tag{13}$$

$$(0 \not\perp\!\!\!\perp 3 \mid 2) \tag{14}$$

$$(0 \perp\!\!\!\perp 3 \mid \{1, 2\}) \tag{15}$$

$$(0 \perp\!\!\!\perp 1 \mid 2) \wedge (0 \not\perp\!\!\!\perp 3 \mid 2) \implies (0 \not\perp\!\!\!\perp 3 \mid \{1, 2\}). \tag{16}$$

Following the IKB construction procedure described in the previous section, propositions (13), (14) and (15) correspond to the following arguments, respectively:

$$\begin{gathered}\left(\left\{(0 \perp\!\!\!\perp 1 \mid 2)\right\}, (0 \perp\!\!\!\perp 1 \mid 2)\right) \\ \left(\left\{(0 \not\perp\!\!\!\perp 3 \mid 2)\right\}, (0 \not\perp\!\!\!\perp 3 \mid 2)\right) \\ \left(\left\{(0 \perp\!\!\!\perp 3 \mid \{1,2\})\right\}, (0 \perp\!\!\!\perp 3 \mid \{1,2\})\right)\end{gathered} \tag{17}$$

while rule (16) corresponds to the argument

$$\left(\left\{(0 \perp\!\!\!\perp 1 \mid 2), (0 \not\perp\!\!\!\perp 3 \mid 2)\right\}, (0 \not\perp\!\!\!\perp 3 \mid \{1,2\})\right). \tag{18}$$

Let us extend this IKB with the following preference values for its propositions and rule.

$$\begin{aligned} \mathit{Pref}[(0 \perp\!\!\!\perp 1 \mid 2)] &= 0.8 \\ \mathit{Pref}[(0 \not\perp\!\!\!\perp 3 \mid 2)] &= 0.7 \\ \mathit{Pref}[(0 \perp\!\!\!\perp 3 \mid \{1,2\})] &= 0.5. \end{aligned}$$

According to Definition (10), the preference of each argument $(\{\sigma\}, \sigma)$ is equal to the preference value of $\{\sigma\}$ which is equal to the preference of σ, as it contains only a single proposition. Thus,

$$\begin{aligned} \mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 1 \mid 2)\right\}, (0 \perp\!\!\!\perp 1 \mid 2)\right)\right] &= 0.8 \\ \mathit{Pref}\left[\left(\left\{(0 \not\perp\!\!\!\perp 3 \mid 2)\right\}, (0 \not\perp\!\!\!\perp 3 \mid 2)\right)\right] &= 0.7 \\ \mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 3 \mid \{1,2\})\right\}, (0 \perp\!\!\!\perp 3 \mid \{1,2\})\right)\right] &= 0.5. \end{aligned}$$

The preference of argument (18) equals the preference of the set of its antecedents, which, according to Eq. (12), is equal to the product of their individual preferences, that is,

$$\mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 1 \mid 2), (0 \not\perp\!\!\!\perp 3 \mid 2)\right\}, (0 \not\perp\!\!\!\perp 3 \mid \{1,2\})\right)\right] = 0.8 \times 0.7 = 0.56.$$

Proposition (15) and rule (16) contradict each other logically, that is, their corresponding arguments (17) and (18) defeat each other. However, argument (18) is not attacked as its preference is 0.56 *which is larger than* 0.5*, the preference of its defeater argument (17). Since no other argument defeats (18), it is acceptable, and (17), being attacked by an acceptable argument, must be rejected. We therefore see that using preferences the inconsistency of Example 1 has been resolved in favor of rule (16).*

Let us now illustrate the defend relation, that is, how an argument can be defended by some other argument. The example also illustrates an alternative resolution for the inconsistency of Example 1, this time in favor of the independence proposition (15).

Example 4 *Let us extend the IKB of Example 3 with two additional independence propositions and an additional rule. The new propositions and their preference are:*

$$\begin{aligned} \mathit{Pref}[(0 \perp\!\!\!\perp 4 \mid \{2,3\})] &= 0.9 \\ \mathit{Pref}[(0 \perp\!\!\!\perp 3 \mid \{2,4\})] &= 0.8 \end{aligned}$$

and the new rule is:

$$(0 \perp\!\!\!\perp 4 \mid \{2,3\}) \wedge (0 \perp\!\!\!\perp 3 \mid \{2,4\}) \quad \Longrightarrow \quad (0 \perp\!\!\!\perp 3 \mid 2).$$

This rule is an instance of the Intersection axiom followed by Decomposition.

The corresponding arguments and preferences are:

$$\begin{aligned} \mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 4 \mid \{2,3\})\right\}, (0 \perp\!\!\!\perp 4 \mid \{2,3\})\right)\right] &= 0.9 \\ \mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 3 \mid \{2,4\})\right\}, (0 \perp\!\!\!\perp 3 \mid \{2,4\})\right)\right] &= 0.8 \end{aligned}$$

corresponding to the two propositions, and

$$\mathit{Pref}\left[\left(\left\{(0 \perp\!\!\!\perp 4 \mid \{2,3\}), (0 \perp\!\!\!\perp 3 \mid \{2,4\})\right\}, (0 \perp\!\!\!\perp 3 \mid 2)\right)\right] = 0.9 \times 0.8 = 0.72 \qquad (19)$$

corresponding to the rule.

As in Example 3, argument (17) is attacked by argument (18). Let us represent this graphically using an arrow from argument a to argument b to denote that a attacks b, that is,

$$\mathit{Argument}\ (18) \longrightarrow \mathit{Argument}\ (17).$$

If the IKB was as in Example 3, (18) would had been accepted and (17) would have been rejected. However, the additional argument (19) now defeats (undercuts) (18) by logically contradicting its antecedent $(0 \not\perp\!\!\!\perp 3 \mid 2)$*. Since the preference of (19), namely* 0.72*, is larger than that of (18), namely* 0.56*, (19) attacks (18). Therefore, (19) defends all arguments that are attacked by argument (18), and in particular (17). Graphically,*

$$\mathit{Argument}\ (19) \longrightarrow \mathit{Argument}\ (18) \longrightarrow \mathit{Argument}\ (17).$$

Note this is not sufficient for accepting (17) as it has not been proved that its defender (19) is itself acceptable. We leave the proof of this as an exercise for the reader.

4. The Argumentative Independence Test (AIT)

The independence-based preference argumentation framework described in the previous section provides a semantics for the acceptance of arguments consisting of independence propositions. However, what we need is a procedure for a test of independence that, given as input a triplet $\sigma = (X, Y \mid \mathbf{Z})$ responds whether X is independent or dependent of Y given $\mathbf{Z}$. In other words, we need a semantics for the acceptance of propositions, not arguments. Let us consider the two propositions related to the input triplet $\sigma = (X, Y \mid \mathbf{Z})$, proposition $(\sigma = \texttt{true})$, abbreviated $\sigma_{\texttt{t}}$, and proposition $(\sigma = \texttt{false})$, abbreviated $\sigma_{\texttt{f}}$, that correspond to independence $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ and dependence $(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})$ of σ, respectively. The basic idea for deciding on the independence or dependence of input triplet σ is to define a semantics for the acceptance or rejection of propositions $\sigma_{\texttt{t}}$ and $\sigma_{\texttt{f}}$ based on the acceptance or rejection of their respective propositional arguments $(\{\sigma_{\texttt{t}}\}, \sigma_{\texttt{t}})$ and $(\{\sigma_{\texttt{f}}\}, \sigma_{\texttt{f}})$. Formally,

$$\begin{aligned} (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \quad &\text{iff} \quad (\{(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})\}, (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})) \text{ is accepted, and} \\ (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \quad &\text{iff} \quad (\{(X \perp\!\!\!\perp Y \mid \mathbf{Z})\}, (X \perp\!\!\!\perp Y \mid \mathbf{Z})) \text{ is accepted.} \end{aligned} \qquad (20)$$

Based on this semantics over propositions, we decide on the dependence or independence of triplet σ as follows:

$$\begin{aligned} \sigma_{\mathtt{t}} = (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \quad &\Longrightarrow \quad (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \\ \sigma_{\mathtt{f}} = (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \quad &\Longrightarrow \quad (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}). \end{aligned} \tag{21}$$

We call the test that determines independence in this manner the **Argumentative Independence Test** or **AIT**. For the above semantics to be well-defined, a triplet σ must be either independent or dependent, that is, not both or neither. For that, exactly one of the antecedents of the above implications of Eq. (21) must be true. Formally,

Theorem 3 *For any input triplet $\sigma = (X, Y \mid \mathbf{Z})$, the argumentative independence test (**AIT**) defined by Eqs. (20) and (21) produces a non-ambiguous decision, that is, it decides σ evaluates to either independence or dependence, but nor both or neither.*

For that to happen, one and only one of its corresponding propositions $\sigma_{\mathtt{t}}$ or $\sigma_{\mathtt{f}}$ must be accepted. A necessary condition for this is given by the following theorem.

Theorem 4 *Given a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with a strict and transitive preference relation π, every propositional argument $(\{\sigma_{\mathtt{t}}\}, \sigma_{\mathtt{t}}) \in \mathcal{A}$ and its negation $(\{\sigma_{\mathtt{f}}\}, \sigma_{\mathtt{f}})$ satisfy*

$$(\{\sigma_{\mathtt{t}}\}, \sigma_{\mathtt{t}}) \textit{ is accepted iff } (\{\sigma_{\mathtt{f}}\}, \sigma_{\mathtt{f}}) \textit{ is rejected.}$$

The above theorem is not sufficient because the propositions may still be in abeyance, but this possibility is ruled out for strict preference relations by Theorem 5, presented in the next section.

The formal proofs of Theorems 3, 4 and 5 are presented in Appendix B. We now illustrate the use of AIT with an example.

Example 5 *We consider an extension of Example 3 to illustrate the use of the AIT to decide on the independence or dependence of input triplet $(0, 3 \mid \{1, 2\})$. According to Eq. (20) the decision depends on the status of the two propositional arguments:*

$$(\{(0 \not\perp\!\!\!\perp 3 \mid \{1,2\})\}, (0 \not\perp\!\!\!\perp 3 \mid \{1,2\})), \textit{ and} \tag{22}$$

$$(\{(0 \perp\!\!\!\perp 3 \mid \{1,2\})\}, (0 \perp\!\!\!\perp 3 \mid \{1,2\})). \tag{23}$$

Argument (23) is equal to argument (17) of Example 3 that was proved to be rejected in that example. Therefore, according to Theorem 4, its negated propositional argument Eq. (22) must be accepted, and we can conclude that triplet $(0, 3 \mid \{1, 2\})$ corresponds to a dependence, that is, we conclude that $(0 \not\perp\!\!\!\perp 3 \mid \{1, 2\})$.

5. The Top-down AIT Algorithm

We now discuss in more detail the top-down algorithm which is used to implement the argumentative independence test, introduced in Section 3. We start by simplifying the recursion of Eq. (7) that determines the state (accepted, rejected, or in abeyance) of an argument a. We then explain the algorithm and analyze its computability (i.e., prove that its recursive execution is always finite) and its time complexity.

To simplify the recursion Eq. (7) we use the following theorem (proved in Appendix B).

Theorem 5 *Let $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ be a PAF with a strict preference relation π. Then no argument $a \in \mathcal{A}$ is in abeyance.*

This theorem reduces the number of states of each argument to two, that is, an argument can be either accepted or not accepted (rejected). We will use the name of the argument a to denote the predicate "a is accepted" and its negation $\neg a$ to denote the predicate "a is rejected." With this notation, the above theorem, and the fact that we have extended the semantics of acceptability from the defeat to the attack relation (using preferences), the recursion of Eq. (7) can be expressed as follows

$$\begin{aligned} a &\iff \forall b \in attackers(a),\ \neg b \\ \neg a &\iff \exists b \in attackers(a),\ b \end{aligned}$$

or, equivalently,

$$\begin{aligned} a &\iff \bigwedge_{b \in attackers(a)} \neg b \\ \neg a &\iff \bigvee_{b \in attackers(a)} b. \end{aligned}$$

Finally, we notice that the second formula is logically equivalent to the first (simply negating both sides of the double implication recovers the first). Therefore, the Boolean value of the dialog tree for a can be computed by the simple expression

$$a \iff \bigwedge_{b \in attackers(a)} \neg b. \tag{24}$$

To illustrate, consider an attacker b of a. If b is rejected, that is, $\neg b$, the conjunction on the right cannot be determined without examining the other attackers of a. Only when all attackers of a are known to be rejected can the value of a be determined, that is, accepted. Instead, if b is accepted, that is, b, the state of $\neg b$ is `false` and the conjunction can be immediately evaluated to `false`, that is, a is rejected regardless of the acceptability of any other attackers.

An iterative version of the top-down algorithm is shown in Algorithm 3. We assume that the algorithm can access a global PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$, with arguments in $\mathcal{A}$ defined over a knowledge base $\mathcal{K} = \langle \Sigma, \Psi \rangle$. Given as input a triplet $t = (X, Y \mid \mathbf{Z})$, if the algorithm returns `true` (`false`) then we conclude that t is independent (dependent). It starts by creating a root node u for the propositional argument U of proposition $t =$ `true` (lines 1–6). According to Eqs. (20) and (21), the algorithm then decides `true` if U is accepted (line 25). Otherwise, the algorithm returns `false` (line 26). This is because in this case, according to Theorem 4, the negation of propositional argument U must be accepted.

Algorithm 3 is an iterative version of a tree traversal algorithm. It maintains a queue of the nodes that have not been expanded yet. A node is expanded when its children are added to the tree. In the algorithm, this is done in the loop of lines 18 to 23, which uses subroutine *getAttackers* of Algorithm 5 to obtain all attackers of an argument. This subroutine finds all attackers of the input argument a in a backward-chaining fashion, that is, given an argument $a = (H, h)$, it searches for all rules in the knowledge base $\mathcal{K}$ whose consequent matches the negation of some proposition in the support H (undercutters), or the negation of its head h (rebutters). Every node maintains a three-valued state variable $STATE \in \{$`nil`, `accepted`, `rejected`$\}$. The `nil` state denotes

Algorithm 3: *independent*(*triplet t*).

1: $f_{\texttt{true}} \leftarrow$ proposition $(t = \texttt{true})$ /* *Creates independence proposition* $(t = \texttt{true})$. */
2: $U_{\texttt{true}} \leftarrow (\{f_{\texttt{true}}\}, f_{\texttt{true}})$
3: $u_{\texttt{true}} \leftarrow$ node for argument $U_{\texttt{true}}$
4: $u_{\texttt{true}}.parent \leftarrow \texttt{nil}$
5: $u.STATE \leftarrow \texttt{nil}$
6: $fringe \leftarrow [u]$ /* *Initialize with u (root).* */
7: {Create global rejected node, denoted by ρ.}
8: $\rho \leftarrow$ node with no argument and state `rejected`
9: **while** $fringe \neq \varnothing$ **do**
10: $u \leftarrow dequeue(fringe)$
11: $attackers \leftarrow getAttackers(u.argument)$
12: **if** $(attackers = \varnothing)$ **then**
13: $u.STATE \leftarrow \texttt{accepted}$
14: **if** $sendMsg(\rho, u) = terminate$ **then** break
15: **end if**
16: $attackers \leftarrow$ sort attackers in decreasing order of preference.
17: {Enqueue attackers after decomposing them.}
18: **for each** $A \in attackers$ **do**
19: $a \leftarrow$ node for argument A
20: $a.parent \leftarrow u$
21: $a.STATE \leftarrow \texttt{nil}$
22: *enqueue a* in $fringe$ /* *See details in text.* */
23: **end for**
24: **end while**
25: **if** $(u.STATE = \texttt{accepted})$ **then return** `true`
26: **if** $(u.STATE = \texttt{rejected})$ **then return** `false`

Algorithm 4: *sendMsg*(*Node c*, *Node p*).

1: {Try to evaluate node p given new information in $c.STATE$}
2: **if** $p \neq \texttt{nil}$ **then**
3: **if** $c.STATE = \texttt{accepted}$ **then** $p.STATE \leftarrow \texttt{rejected}$
4: **else if** ($\forall$ children q of p, $q.STATE \neq \texttt{rejected}$) **then** $p.STATE \leftarrow \texttt{accepted}$
5: {If p was successfully evaluated, try to evaluate its parent by sending message upward.}
6: **if** $p.STATE \neq \texttt{nil}$ **then**
7: **return** $sendMsg(p, p.parent)$
8: **else**
9: **return** *continue*
10: **end if**
11: **else**
12: **return** *terminate* {The root node has been evaluated.}
13: **end if**

that the value of the node is not yet known, and a node is initialized to this state when it is added to the tree.

The algorithm recurses down the dialog tree until a node is found that has no attackers (line 12). Such a node is accepted in line 13, that is, the conjunction of Eq. (24) is trivially true, and its *STATE* is propagated upwards toward the root to the

Algorithm 5: Finds all attackers of input argument a in knowledge base $\mathcal{K} = \langle \Sigma, \Psi \rangle$: $getAttackers(a = (H, h))$

1: $attackers \leftarrow \varnothing$
2: {Get all undercutters or rebutters of a.}
3: **for all** propositions $\varphi \in H \cup \{h\}$ **do**
4: {Get all defeaters of proposition φ.}
5: **for all** rules $(\Phi_1 \wedge \Phi_2 \ldots \wedge \Phi_n \implies \neg\varphi) \in \Psi$ **do**
6: {Find all propositionalizations of the rule whose consequent matches $\neg\varphi$.}
7: **for all** subsets $\{\varphi_1, \varphi_2 \ldots, \varphi_n\}$ of Σ s.t. $\Phi_1 \equiv \varphi_1, \Phi_2 \equiv \varphi_2 \ldots \Phi_n \equiv \varphi_n$ **do**
8: $d \leftarrow (\{\varphi_1 \wedge \varphi_2 \ldots \varphi_n\}, \neg\varphi)$ {Create defeater.}
9: {Is the defeater an attacker?}
10: **if** $\neg(a \gg_\pi d)$ **then**
11: $attackers \leftarrow attackers \cup \{d\}$
12: **end if**
13: **end for**
14: **end for**
15: **end for**
16: **return** *attackers*

parent using subroutine *sendMsg* (Algorithm 4). Every time a node receives a message from a child, if the message is `accepted`, the node is rejected (line 3 of Algorithm 4), otherwise the node is accepted if all its children has been evaluated to `rejected` (line 4 of Algorithm 4). The subroutine *sendMsg* then proceeds recursively by forwarding a message to the parent whenever a node has been evaluated (line 7). If the root is reached and evaluated, the message is sent to its parent, which is `nil`. In this case, the subroutine returns the special keyword *terminate* back to the caller, indicating that the root has been evaluated and thus the main algorithm (Algorithm 3) can terminate. The caller can be either the subroutine *sendMsg*, in which case it pushes the returned message up the method-calling stack, or the top-down algorithm in line 14, in which case its "while" loop is terminated.

An important part of the algorithm is yet underspecified, namely the order in which the attackers of a node are explored in the tree (i.e., the priority with which nodes are enqueued in line 22). This affects the order of expansion of the nodes in the dialog tree. Possible orderings are depth-first, breadth-first, iterative deepening, as well as informed searches such as best-first when a heuristic is available. In our experiments we used iterative deepening because it combines the benefits of depth-first and breadth-first search, that is, small memory requirements on the same order as depth-first search (i.e., on the order of the maximum number of children a node can have) but also the advantage of finding the shallowest solution like breadth-first search. We also used a heuristic for enqueuing the children of a node. According to iterative deepening, the position in the queue of the children of a node is specified relative to other nodes, but not relative to each other. We therefore specified the relative order of the children according to the value of the preference function: children with higher preference are enqueued first (line 16 of the top-down algorithm), and thus, according to iterative deepening, would be dequeued first.

5.1. Computability of the Top-Down Algorithm

An open question is whether the top-down algorithm is computable, that is, whether it always terminates. In this section we prove that it is. To prove this we need to show that under certain general conditions the acceptability of an argument a can always be determined, that is, that the algorithm always terminates. This is proved by the following theorem.

Theorem 6 *Given an arbitrary triplet $t = (X, Y \mid \mathbf{Z})$, and a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with a strict preference relation π, Algorithm 3 with input t over $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ terminates.*

The proof consists on showing that the path from the root a to any leaf is always finite. For that, the concept of an attack sequence is needed.

Definition 16 *An* attack sequence *is a sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of n arguments such that for every $2 \leq i \leq n$, a_i attacks a_{i-1}.*

By the manner in which the top-down algorithm constructs the dialog tree it is clear that any path from the root to a leaf is an attack sequence. It therefore suffices to show that any such sequence is finite. This is done by the following theorem.

Theorem 7 *Every attack sequence $\langle a_1, a_2, \ldots, a_n \rangle$ in a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with strict π and finite $\mathcal{A}$ is finite.*

Intuitively, if the preference relation is strict then an element can attack its predecessor in the sequence but not vice versa. Since the set of arguments $\mathcal{A}$ is finite, the only way for an attack sequence to be infinite is to contain a cycle. In that case, an argument would be attacking at least one of its predecessors, which cannot happen in a PAF with a strict preference relation. We present formal proofs of Theorems 6 and 7 in Appendix A.

We thus arrived at the important conclusion that, under a strict preference function and a finite argument set, the state of any argument is computable. As we showed in Section 3.3, the preference function for independence knowledge bases is strict, and thus the computability of the top-down algorithm is guaranteed.

5.2. Computational Complexity of the Top-Down Algorithm

Since Algorithm 3 is a tree traversal algorithm, its time complexity can be obtained by techniques contained in standard algorithmic texts, for example, Cormen et al. (2001). The actual performance depends on the tree exploration procedure. In our case we used iterative deepening due to its linear memory requirements in d, where d is the smallest depth at which the algorithm terminates. Iterative deepening has a worst-time time complexity of $O(b^d)$, where b is an upper bound on the *dialog tree branching factor*. Therefore, for a constant $b > 1$ the execution time is exponential in d in the worst case. Furthermore, for the case of independence tests, b itself may also be exponential in n (the number of variables in the domain). This is because the inference rules of Eqs. (5) and (6) are universally quantified, and therefore their propositionalization (lines 7–13 of Algorithm 5), may result in an exponential number of rules with the same consequent (attackers). A tighter bound may be possible but, lacking such a bound, we introduce in the next section an approximate top-down algorithm, which reduces the running time to polynomial. As we show in our experiments, the use of this approximation does not appreciably affect the accuracy improvement due to argumentation.

6. The Approximate Top-Down AIT Algorithm

As the top-down algorithm has a theoretically exponential running time in the worst case, we hereby present a practical polynomial-time approximation of the top-down algorithm. We make use of two approximations: (a) To address the exponential behavior due to the depth of the dialog tree we apply a *cutoff depth d* for the process of iterative deepening. (b) To address the potentially exponential size of the branching factor b (which equals the maximum number of defeaters of any argument appearing in the dialog tree) we limit the number of defeaters of each node—thus bounding the number of its attackers/children—to a polynomial function of n (the domain size) during the propositionalization process of Algorithm 5 (lines 7–13). Let (H, h) be an argument and let $\varphi \in H \cup \{h\}$ be one of its propositions, as in line 3 of Algorithm 5. The set of attackers Σ_φ of (H, h) consists of all rules $\{\varphi_1 \wedge \varphi_2 \ldots \wedge \varphi_k \implies \neg\varphi\}$ of Σ, for some constant upper bound k on the size of their support. If $\varphi = (\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$ and $\varphi_i = (\mathbf{X}_i, \mathbf{Y}_i \mid \mathbf{Z}_i)$ for all $1 \leq i \leq k$, then our approximation generates and uses a subset of Σ_φ in the dialog tree such that

$$\begin{array}{ccccc} |\mathbf{X}| - c & \leq & |\mathbf{X}_i| & \leq & |\mathbf{X}| + c \\ |\mathbf{Y}| - c & \leq & |\mathbf{Y}_i| & \leq & |\mathbf{Y}| + c \\ |\mathbf{Z}| - c & \leq & |\mathbf{Z}_i| & \leq & |\mathbf{Z}| + c \end{array} \tag{25}$$

where $|\cdot|$ denotes set cardinality, and c is a user-specified integer parameter that defines the approximation. We call this the **approximate top-down algorithm**. The computational complexity of the approximate top-down algorithm is polynomial in n, as shown in the next section.

6.1. Test Complexity of the Approximate Top-Down Algorithm

In this section we prove that the number of statistical tests required by the Approximate Top-Down algorithm is polynomial in n. As described in the previous section, the approximate algorithm generates a bounded number of attackers for each proposition in the argument corresponding to some node in the dialog tree. A bound on the number of the possible attackers can be defined by the approximation of Eq. (25). These equations dictate that the size of each possible set $\mathbf{X}_i$ in some proposition $(\mathbf{X}_i, \mathbf{Y}_i \mid \mathbf{Z}_i)$ of some attacker of proposition $(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$ is between $|\mathbf{X}| + c$ and $|\mathbf{X}| - c$ (inclusively). As the number of elements that can be members of $\mathbf{X}_i$ is bounded by n (the domain size), this produces at most n^{2c+1} possible instantiations for set $\mathbf{X}_i$. Similarly, the number of possible instantiations for $\mathbf{Y}_i$ and $\mathbf{Z}_i$ is also n^{2c+1}. Therefore, an upper bound for the number of matches to some proposition in the antecedent of an attacking rule is $O(n^{6c+3})$ for some constant c. As there are r rules in the rule set and up to k propositions in each rule for some constants r and k (for example, $r = 5$ and $k = 3$ for Eq. (5) and $r = 8$ and $k = 4$ for Eq. (6)), an upper bound on the number of children of a node in the dialog tree is $O(rkn^{6c+3})$, and thus an upper bound on the number of nodes in the dialog tree of depth d is $O((rk)^d n^{d(6c+3)})$. As we demonstrate in our experiments, this is a rather loose upper bound and the performance of the approximate top-down algorithm is reasonable in practice, but it does serve to show that the theoretical worst-case performance is polynomial in n. In the experiments shown in the next section we used $c = 1$ and $d = 3$.

7. Experimental Results

We conducted experiments on sampled and real-world data sets for the purpose of (a) evaluating the accuracy improvement of the argumentative test (both the exact and approximate versions) over its statistical counterpart; (b) demonstrating the performance improvements that can be achieved by the approximate version compared to the exact counterpart, without significant reduction in accuracy improvement; and (c) evaluating the improvements that result by the use of the argumentative framework for causal discovery. We address these issues below.

7.1. Comparative Evaluation of Bottom-Up, Exact Top-Down, and Approximate Top-Down Argumentative Tests

In this section we demonstrate that the argumentation approach, implemented either by the (exact) bottom-up or the exact top-down algorithm (Algorithm 3), improves the accuracy of independence tests on small data sets. We also show that the approximate top-down algorithm (see Section 6) has accuracy performance improvements similar to its exact counterpart but significantly better execution times (orders of magnitude), that make it more practical and usable for larger domains. As the output of the bottom-up algorithm is guaranteed to be equal to the exact top-down algorithm as Theorem 2 of Section 3, we omit accuracy results for the bottom-up algorithm here.

As the exact algorithm is impractical for large domains, for the present comparison we sampled data sets from two randomly generated Bayesian networks with $n = 8$ nodes. The networks were generated using *BNGenerator* (Ide et al., 2002), a publicly available Java package, with maximum degree per node τ equal to 3 and 7 to evaluate the performance in sparsely as well as densely connected domains. A large data set D was sampled from each network and our experiments were conducted on subsets of it containing an increasing number of data points N. This was done in order to assess the accuracy on varying conditions of reliability, as the reliability of a test varies (typically increases) with the amount of data available. To reduce variance, each experiment was repeated for ten data subsets of equal size, obtained by permuting the data points of D randomly and using the first N of them as input to our algorithms.

We first compare the accuracy of argumentative tests versus their purely statistical counterparts (i.e., the G^2 test) on several data sets sampled from randomly generated Bayesian networks. Sampled data experiments have the advantage of a more precise estimation of the accuracy since the underlying model is known. We present experiments for two versions of the exact top-down argumentative test, one using Pearl's general axioms of of Eq. (5), denoted $\mathbf{AIT_t}$**-G**, and another that uses Pearl's "directed" axioms of Eq. (6), denoted $\mathbf{AIT_t}$**-D**, as well as two versions of the approximate top-down argumentative test, denoted $\widehat{\mathbf{AIT}}_\mathbf{t}$**-G** and $\widehat{\mathbf{AIT}}_\mathbf{t}$**-D** respectively. We abbreviate purely statistical independence tests as **SIT**.

We report the estimated accuracy, which, for each data set, is calculated by comparing the result of a number of conditional independence tests (SITs or AITs) on data with the true value of independence, computed by querying the underlying model for the conditional independence of the same test using d-separation. Since the number of possible tests is exponential, we estimated the independence accuracy by randomly sampling a set $\mathcal{T}$ of 1,000 triplets $(X, Y, \mathbf{Z})$, evenly distributed among all possible conditioning set sizes $m \in \{0, \ldots, n-2\}$, that is, $1000/(n-1)$ tests for each m. The independence or dependence value of the triplets (in the true, underlying model) were not controlled, but left to be decided randomly. This resulted in a non-uniform distribution of dependencies

and independences. For instance, in the experiments shown next ($n = 8$, $\tau = 3, 7$), the average proportion of independences vs. dependencies was 36.6% to 63.4% respectively for $\tau = 3$, and 11.4% to 88.6% respectively for $\tau = 7$. Denoting a triplet in $\mathcal{T}$ by t, by $I_{\text{true}}(t)$ the result of a test on t performed on the underlying model, and by $I_{\text{data-}\mathcal{Y}}(t)$ the results of performing a test on t of type $\mathcal{Y}$ on data, for $\mathcal{Y}$ equal to SIT, AIT_t-G, AIT_t-D, $\widehat{\text{AIT}}_\text{t}$-G, or $\widehat{\text{AIT}}_\text{t}$-D, the estimated accuracy of test type $\mathcal{Y}$ is defined as

$$\widehat{acc}^{\text{data}}_{\mathcal{Y}} = \frac{1}{|\mathcal{T}|}\left|\left\{t \in \mathcal{T} \mid I_{\text{data-}\mathcal{Y}}(t) = I_{\text{true}}(t)\right\}\right|.$$

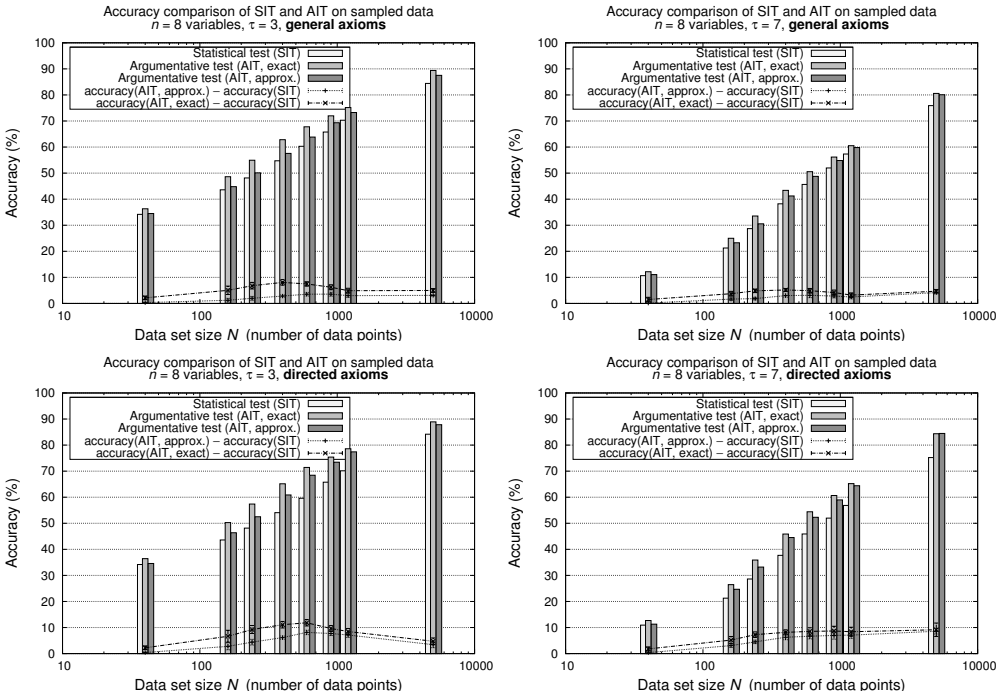


Figure 2: Accuracy comparison of statistical tests (SIT) vs. exact and approximate argumentative tests for domain size $n = 8$ and maximum degree per node $\tau = 3, 7$. The histograms show the absolute value of the accuracy while the line curves show the difference between SIT and the argumentative tests. 95% confidence intervals are also shown for the line graphs. **Top row:** General axioms. **Bottom row:** Directed axioms.

Figure 2 (top row) shows a comparison of the SIT with the exact and approximate top-down argumentative test over the general axioms for data set with increasing number of data points. The figure shows two plots for $\tau = 3, 7$ of the mean values (over runs for ten different data sets) of $\widehat{acc}^{\text{data}}_{\text{SIT}}$, $\widehat{acc}^{\text{data}}_{\text{AIT}_\text{t}\text{-G}}$, and $\widehat{acc}^{\text{data}}_{\widehat{\text{AIT}}_\text{t}\text{-G}}$ (histograms) and the difference between the accuracies of the AIT tests and the statistical one (line graphs) for various data set sizes N. A positive value of the difference corresponds to an improvement of the argumentative test over SIT. The plots also show the statistical significance of this difference with 95% confidence intervals (error bars), that is, the interval

around the mean value that has a 0.95 probability of containing the true difference. The figure demonstrates that there exist modest but consistent and statistically significant improvements in the accuracy of both the exact and approximate argumentative tests over the statistical test. We can observe improvements over the entire range of data set sizes in both cases with maximum improvements of up to 9% and 6% for the exact and approximate cases respectively (at $\tau = 3$ and $N = 600$).

In certain situations where the experimenter knows that the underlying distribution belongs to the class of Bayesian networks, it is appropriate to use the specific axioms of Eq. (6) instead of the general axioms of Eq. (5). The bottom row of Figure 2 presents the same comparison as the top row but for the exact and approximate argumentative tests AIT_t-D and $\widehat{\text{AIT}}_\text{t}$-D that use the directed axioms instead of the general ones. As in the case for AIT using the general axioms, we can observe statistically significant improvements over the entire range of data set sizes in both cases. In this case however, the improvements are larger, with maximum increases in the accuracy of the exact and approximate test of up to 13% and 9% respectively (again for $\tau = 3$ and $N = 600$).

We also evaluated the accuracy of these tests for increasing conditioning set sizes. Figures 3 and 4 show a comparison of the SIT with the exact and approximate top-down argumentative test using the general and directed axioms respectively, for accuracies over increasing conditioning set size for data sizes $N = 160, 900$, and 5000 (different rows). We can observe statistically significant improvements over the entire range of conditioning set sizes in all twelve graphs. Except in one case (directed axioms, $N = 5000$, $\tau = 3$), all graphs show an upward trend in accuracy for increasing conditioning set size, representing a positive impact of the argumentative approach that increases with a decrease in test reliability, that is, increasing conditioning set size.

We also compared the execution times of the bottom-up, exact top-down and approximate top-down algorithms on the same data sets. To run the bottom-up algorithm we generated the set of all propositional arguments possible, that is, arguments of the form $(\{\sigma\}, \sigma)$, by iterating over all possible triplets $(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$, and inserted them in the knowledge base together with their preference, as described in Section 3.1. Similarly, for the set of axioms that we used in each case, that is, either the general (Eq. (5)) or the specific ones (Eq. (6)), we iterated over all possible matches of each rule, inserting the corresponding (single-headed and decomposed) instantiated rule in the knowledge base again together with its preference. The reason for including all propositional and rule-based arguments in our IKB is to allow the argumentation framework to consider all possible arguments in favor of or against an independence query. We compared the bottom-up algorithm AIT_b, the exact top-down algorithms AIT_t, and the approximate top-down algorithm $\widehat{\text{AIT}}_\text{t}$. For this, we measured the time it takes to discover the structure of a Bayesian networks using three versions of the PC algorithm (Spirtes et al., 2000), each using one of the three argumentative tests AIT_b, AIT_t, or $\widehat{\text{AIT}}_\text{t}$ to conduct the independence tests. As usual, we consider two versions of each test AIT_b, AIT_t, and $\widehat{\text{AIT}}_\text{t}$, one that uses the general axioms of Eq. (5), that is, AIT_b-G, AIT_t-G, and $\widehat{\text{AIT}}_\text{t}$-G, respectively, and one that uses the specific axioms of Eq. (6) (applicable to Bayesian networks), that is, AIT_b-D, AIT_t-D, and $\widehat{\text{AIT}}_\text{t}$-D, respectively. The data sets used are the same as the ones used in the accuracy comparisons above.

Figure 5 plots the execution time of argumentative tests AIT_b-G vs. AIT_t-G vs. $\widehat{\text{AIT}}_\text{t}$-G (top row) and AIT_b-D vs. AIT_t-D vs. $\widehat{\text{AIT}}_\text{t}$-D (bottom row) for tests that were conducted by the PC algorithm while learning the structure. Note that both the x and y-axes are plotted in log-scale. We can observe improvements in the execution time

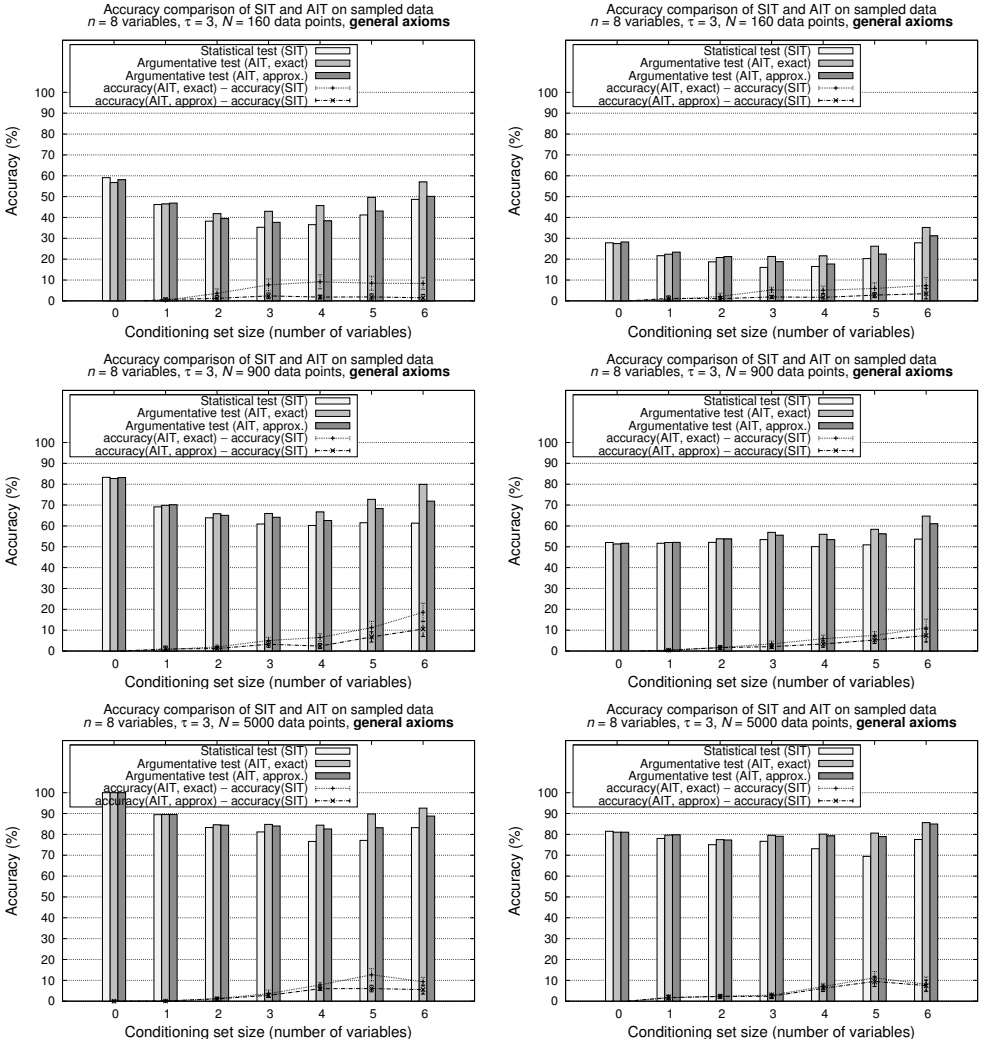


Figure 3: Accuracy comparison of SIT vs. exact (AIT_t-G) and approximate ($\widehat{\text{AIT}}_\text{t}$-G) argumentative tests over the general axioms for increasing conditioning set sizes. The six plots correspond to maximum degrees per node $\tau = 3, 7$, and data set sizes $N = 160, 900$ and 5000.

of the exact top-down algorithm over that of the bottom-up algorithm of an order of magnitude over the entire range of data set sizes in all four plots. We can also see improvement of a similar order between the exact and approximate top-down argumentative algorithms. For instance, for the general axioms and $\tau = 3$ (top-left plot), the execution time for $N = 5000$ is 2749 seconds for the bottom-up against 107 seconds for the exact top-down and 15 seconds for the approximate top-down algorithm. We see even more pronounced execution time improvements when using the directed axioms (bottom row of Fig. 5).

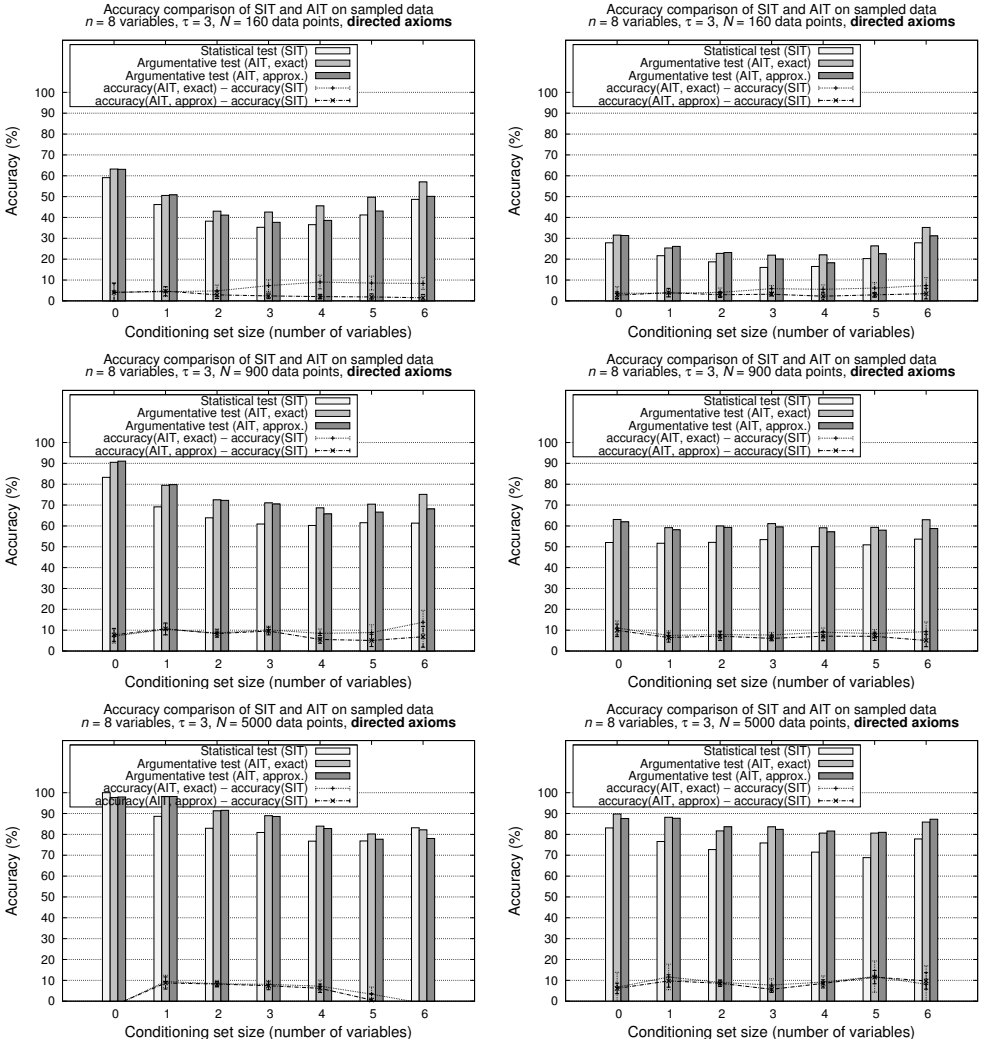


Figure 4: Same as Figure 3 but for AIT using the directed axioms instead of the general ones.

The execution-time results demonstrate that the exact top-down algorithm performs significantly better than the bottom-up algorithm, while producing the exact same output (according to Theorem 2 of Section 3). This implies a clear advantage of using the top-down over the bottom-up algorithm. Furthermore, we also saw that the approximate top-down algorithm performs similarly in terms of accuracy improvement while having polynomial worst-case execution time and in practice being several orders of magnitude faster than the exact top-down algorithm, which is exponential in the worst-case. As in the next two sections we continue our evaluation on domains significantly larger than the $n = 8$ variables that we examined here, it would be difficult or impractical for the exact algorithms to be employed. For these reasons in the following

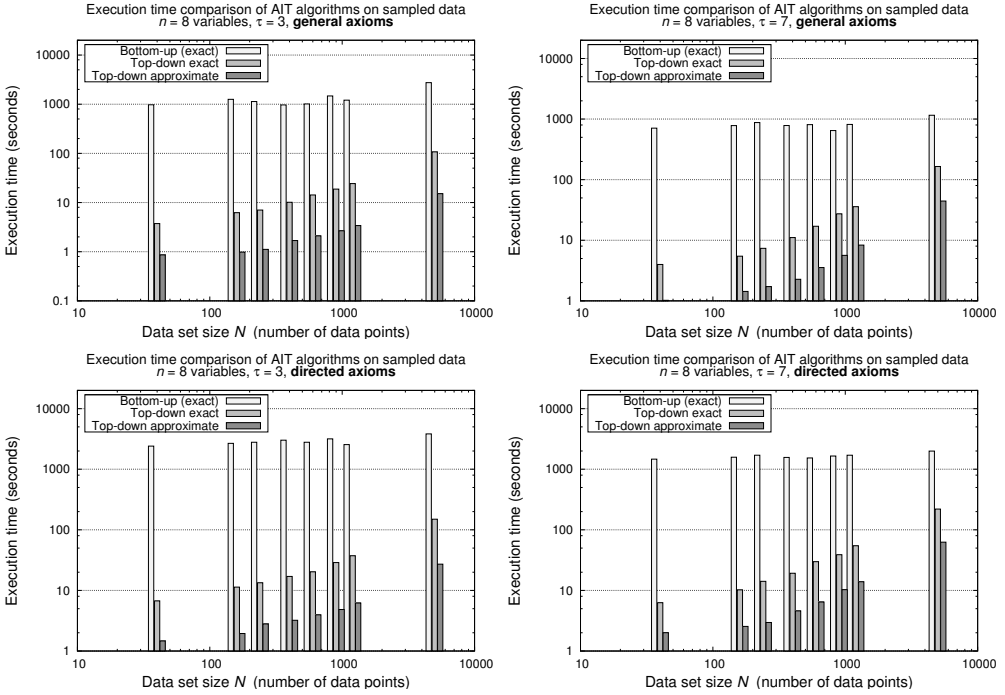


Figure 5: Execution time comparison for the PC algorithm when it uses the bottom-up and exact top-down and approximate top-down argumentative tests to learn the structure of a Bayesian network from data sampled from Bayesian models with domain size $n = 8$, maximum degrees $\tau = 3, 7$. The bars show the absolute value of the running time using a logarithmic scale. **Top row:** general axioms. **Bottom row:** directed axioms.

experiments we use the more practical approximate algorithm, which can be applied to larger domains.

7.2. Causal Discovery in Larger Domains

We also conducted experiments that demonstrate the performance of the approximate top-down algorithm by (a) showing its applicability to large domains, and (b) demonstrating positive improvements in accuracy of argumentative tests on the learning of the structure of Bayesian networks, the main problem faced by causal discovery algorithms. In the following experiments we used the PC algorithm. We compared the true structure of the underlying model to the resulting structure of the PC algorithm when it uses SITs as independence tests, denoted PC-SIT, and its output when it uses argumentative independence tests, denoted PC-$\widehat{\text{AIT}}_t$-D, when using the directed axioms.

We evaluated the resulting networks by their ability to accurately represent the true independences in the domain, calculated by comparing the results (`true` or `false`) of a number of conditional tests conducted using d-separation on the output networks (PC-SIT or PC-$\widehat{\text{AIT}}_t$-D). Denoting by $\mathcal{T}$ this set of 2,000 triplets, by $t \in \mathcal{T}$ a triplet, by $I_{\text{true}}(t)$ the result of a test performed on the underlying model, and by $I_{\text{PC-}\mathcal{Y}}(t)$ the result

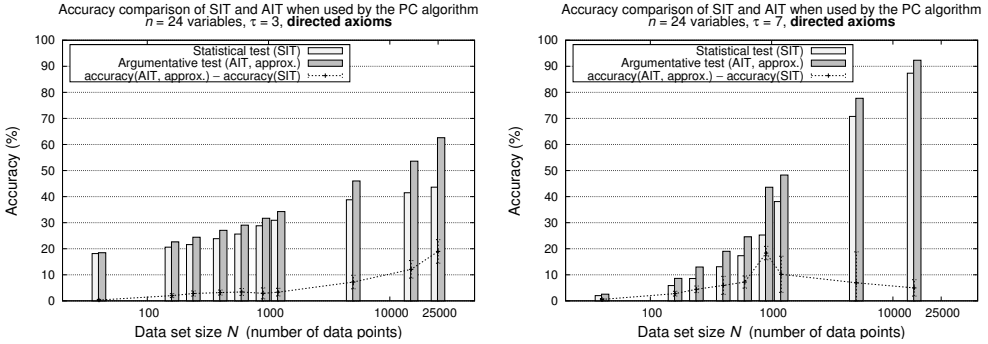


Figure 6: Comparison of statistical tests (SIT) vs. approximate argumentative tests on the directed axioms ($\widehat{\text{AIT}}_t$-D) for data sets sampled from Bayesian models for domain size $n = 24$ and maximum degrees $\tau = 3, 7$.

of performing a d-separation test on the network output by the PC algorithm using the $\mathcal{Y}$ test, $\mathcal{Y}$ equal to SIT or $\widehat{\text{AIT}}_t$-D, the estimated accuracy is defined as

$$\widehat{acc}^{\text{PC}}_{\mathcal{Y}} = \frac{1}{|\mathcal{T}|}\left|\left\{t \in \mathcal{T} \mid I_{\text{PC-}\mathcal{Y}}(t) = I_{\text{true}}(t)\right\}\right|. \tag{26}$$

We considered data sampled from randomly generated Bayesian networks of sizes $n = 24$, and maximum degrees $\tau = 3, 7$. For each network we sampled ten data sets, and, for each data set, we conducted experiments on subsets of D containing an increasing number of data points. We report the average over the ten data sets of the estimated accuracy calculated using Eq. (26), for $\mathcal{Y} =$ SIT or $\widehat{\text{AIT}}_t$-D, as well as the difference between the average accuracies including the 95% confidence interval for the difference.

Figure 6 shows a comparison of the argumentative tests $\widehat{\text{AIT}}_t$-D using the directed axioms with the corresponding SIT. The figure shows two plots for different values of τ of the mean values (over runs for ten different data sets) of $\widehat{acc}^{\text{PC}}_{\text{SIT}}$ and $\widehat{acc}^{\text{PC}}_{\widehat{\text{AIT}}_t\text{-D}}$ (histograms), the difference between these averages (line graph), and the 95% confidence intervals for the difference (error bars), for different data set sizes N. As usual, a positive value of the difference corresponds to an improvement of $\widehat{\text{AIT}}_t$-D over SIT. As in practically all experiments so far, we have statistically significant improvements over the entire range of data set sizes, with maximum improvements of up to 20% for $\tau = 3$, $N = 25000$, and $\tau = 7$, $N = 900$. The corresponding execution times for the entire PC algorithm are shown in Fig. 7. We can make two observations from this graph. One, the cost is significantly lower for sparse domains, which benefits real-world application domains that are sparse. The second observation is that the execution time scales linearly with the number of data points; this exhibits the same behavior as the use of a SIT test in PC, as each test needs to scan the data set once to compute the contingency table and relevant test statistics.

In summary, these results demonstrate that the approximate argumentative test is practical for larger domains and can result in positive, statistically significant accuracy improvements when used for causal discovery. However, the cost of AIT for large data sets, although not prohibitive, can be non-negligible. Therefore the accuracy benefits of

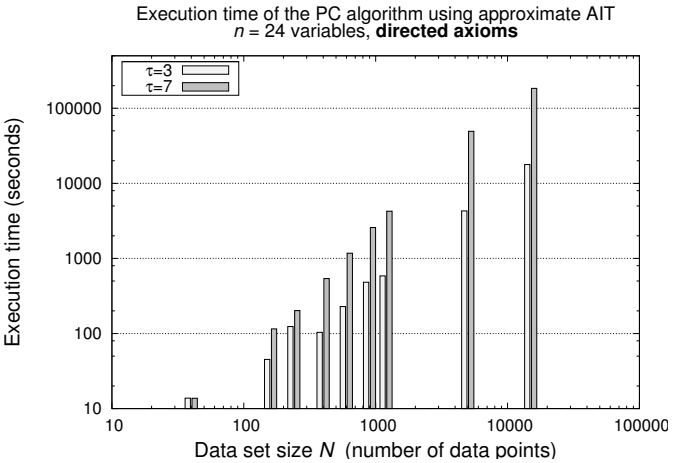


Figure 7: Execution times for the PC algorithm using the approximate argumentative test on the directed axioms ($\widehat{\text{AIT}}_t$-D) on data sets sampled from Bayesian models for domain size $n = 24$ and maximum degrees $\tau = 3, 7$. For the approximate AIT test we limited the depth of the dialog tree to 3 and its the branching factor as described in Section 6.

AIT vs. a SIT must be carefully weighed off the ability of the user to expend the extra computation. Note that the practicality of the approximate algorithm also depends on the parameters used (the cutoff depth of iterative deepening and the branching factor limit—see Section 6); different parameter values or alternative ways of limiting the size of the dialog tree may be needed for even larger domains.

7.3. Real-world and Benchmark Data Experiments

While the sampled data set studies of the previous section have the advantage of a more controlled and systematic study of the performance of the algorithms, experiments on real-world data are necessary for a more realistic assessment. In this section we present experiments on a number of real-world and benchmark data sets obtained from the UCI machine learning repository (D. J. Newman and Merz, 1998) and the Knowledge Discovery Data repository (Hettich and Bay, 1999). As in the sampled data case of the previous section, for each data set D, we conducted experiments on subsets of D containing an increasing number of data points N to assess the performance of the independence tests on varying conditions of reliability. Again, to reduce variance we repeated each experiment ten times, each time choosing a different randomly selected data subset of equal size.

Because for real-world data sets the underlying model is unknown, we could only be sure the general axioms of Eq. (5) apply. We therefore only used these axioms in this section. Also, as mentioned in the previous section, because some of the data sets have much larger domains (e.g., the alarm data set contains 37 variables), and given the exponential nature of the exact algorithms we could only perform experiments for the approximate version of the argumentative test. For these reasons, in the following experiments we only report the accuracy of $\widehat{\text{AIT}}_t$-G, the approximate argumentative

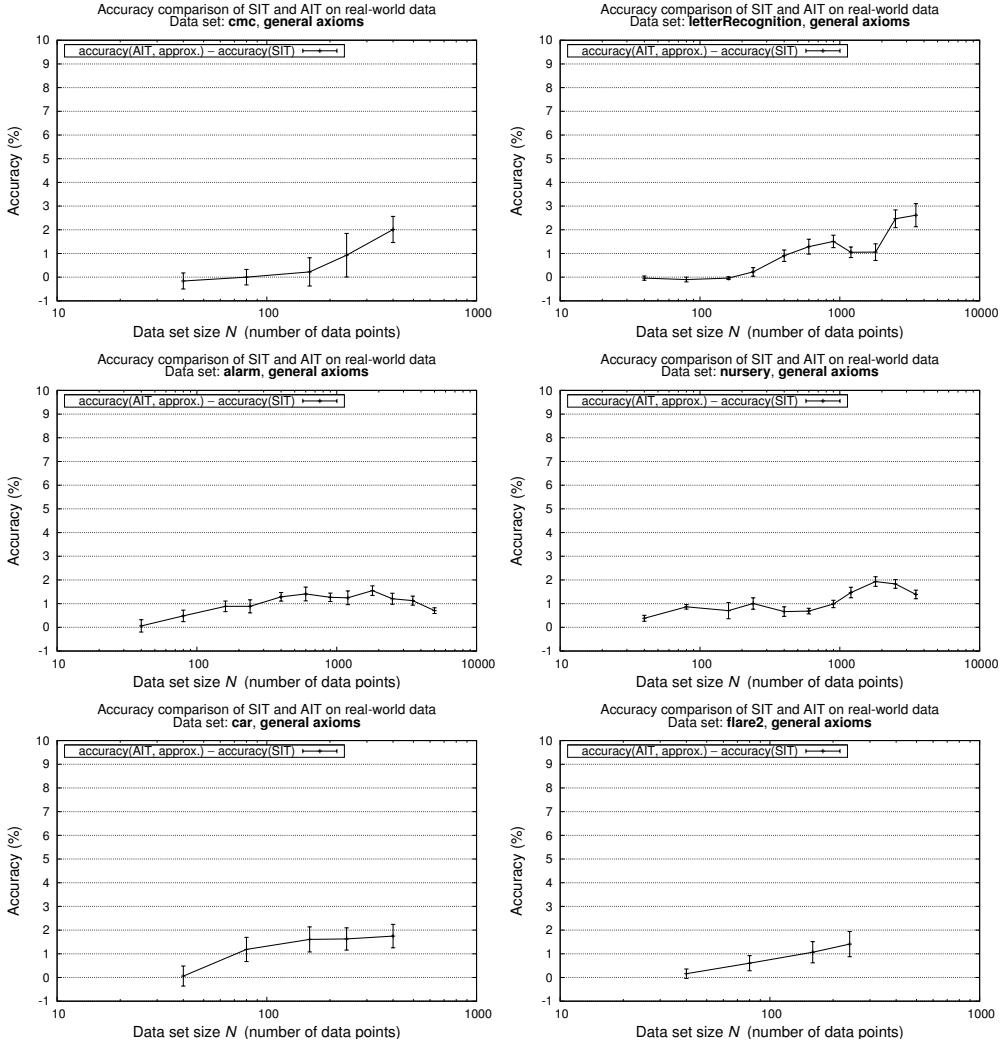


Figure 8: Difference in the mean value of the accuracy $\widehat{\text{AIT}}_t$-G with the mean value of the accuracy of SIT for a number of real-world data sets. The error bars denote the 95% confidence interval of the difference.

independence test defined over the general axioms. Unfortunately, for real-world data the underlying model is typically unknown and therefore it is impossible to know the true value of any independence. We therefore approximate it by a statistical test on the entire data set, and limit the size of the data set subsets that we use up to a third of the size of the entire data set. This corresponds to the hypothetical scenario that a much smaller data set is available to the researcher, allowing us to evaluate the improvement of argumentation under these more challenging situations. Again, as in the previous two sections, for comparison we sampled 2,000 triplets and calculated the accuracy as a fraction of tests correct, where for the true value of independences and dependences we used the method just described.

	Data set Domain size Data set size	|car| 7 1730	|cmc| 10 1475	|flare2| 13 1067	|letterRecognition| 17 20002	|nursery| 9 12962	|alarm| 37 20003
$N = 40$	SIT	80.1	**77.8**	77.0	**47.9**	83.3	76.7
	$\widehat{\text{AIT}}_t$-G	80.1	77.5	**77.1**	47.8	**83.8**	76.7
	$\widehat{\text{AIT}}_t$-G − SIT	0.0 ± 0.7	-0.3 ± 0.6	0.1 ± 0.3	-0.1 ± 0.2	0.4 ± 0.1	0.0 ± 0.4
	Runtime of $\widehat{\text{AIT}}_t$-G (ms)	0.56	1.07	2.61	4.19	0.88	52.06
$N = 240$	SIT	86.7	84.1	85.5	50.7	86.1	84.3
	$\widehat{\text{AIT}}_t$-G	**88.6**	**84.7**	**86.9**	**51.0**	**87.2**	**85.1**
	$\widehat{\text{AIT}}_t$-G − SIT	1.9 ± 0.6	0.5 ± 1.2	1.3 ± 0.8	0.2 ± 0.4	1.1 ± 0.4	0.8 ± 0.3
	Runtime of $\widehat{\text{AIT}}_t$-G (ms)	1.37	5.19	8.73	90.50	1.84	202.05
$N = 600$	SIT				55.8	88.5	88.6
	$\widehat{\text{AIT}}_t$-G				**57.3**	**89.3**	**89.8**
	$\widehat{\text{AIT}}_t$-G − SIT				1.5 ± 0.5	0.8 ± 0.1	1.2 ± 0.4
	Runtime of $\widehat{\text{AIT}}_t$-G (ms)				575.53	4.37	547.77
$N = 1200$	SIT				63.3	89.7	90.8
	$\widehat{\text{AIT}}_t$-G				**64.3**	**91.2**	**92.0**
	$\widehat{\text{AIT}}_t$-G − SIT				1.0 ± 0.3	1.5 ± 0.3	1.2 ± 0.4
	Runtime of $\widehat{\text{AIT}}_t$-G (ms)				2008.76	14.05	1151.05
$N = 3500$	SIT				73.8	94.1	95.2
	$\widehat{\text{AIT}}_t$-G				**76.5**	**95.4**	**96.3**
	$\widehat{\text{AIT}}_t$-G − SIT				2.6 ± 0.7	1.3 ± 0.3	1.1 ± 0.3
	Runtime of $\widehat{\text{AIT}}_t$-G (ms)				24540.51	76.48	3895.2

Table 1: Average accuracies (in percentage) of SIT and $\widehat{\text{AIT}}_t$-G, their differences (denoted $\widehat{\text{AIT}}_t$-G − SIT in the table), the 95% confidence interval for the difference, and the average runtime per test (in ms) for $\widehat{\text{AIT}}_t$-G for several real-world and benchmark data sets. For each data set the table shows these quantities for number of data points $N = 40, 240, 600, 1200, 3500$. The best performing algorithm ($\widehat{\text{AIT}}_t$-G or SIT, with respect to accuracy) is indicated in bold. Empty cells correspond to cases where one third of the data set was smaller than the value of N in that column.

Figure 8 and Table 1 show the result of our comparison between the argumentative test $\widehat{\text{AIT}}_t$-G and statistical test SIT for real-world data sets. In the table, the best-performing method is shown in bold. The figure contains 6 plots, one for each data set, depicting the difference between the mean value of the accuracy of $\widehat{\text{AIT}}_t$-G and that of SIT, where as usual a positive value denotes an improvement of $\widehat{\text{AIT}}_t$-G over SIT. While in a few cases the average difference is negative (e.g., data set |cmc|, $N = 40$), in each case the negative value is not statistically significant as the confidence interval contains a portion of the positive half-plane. The figure demonstrates a clear advantage of the argumentative approach, with all data sets reaching statistically significant positive improvements in accuracy of up to 3% and all confidence intervals covering positive values either partially or completely. The table also shows the average execution time (in ms) for the $\widehat{\text{AIT}}_t$-G tests evaluated.

8. Conclusion

We presented a framework for addressing one of the most important problems of independence-based structure discovery algorithms, namely the problem of unreliability of statistical independence tests. Our main idea was to recognize the existence of interdependences among the outcomes of conditional independence tests—in the form of Pearl's axiomatic characterization of the conditional independence relation—that can

be seen as integrity constraints and exploited to correct unreliable statistical tests. We modeled this setting as a knowledge base containing conditional independences that are potentially inconsistent, and used the preference-based argumentation framework to reason with and resolve these inconsistencies. We presented in detail how to apply the argumentation framework to independence knowledge bases and how to compute the preference among the independence propositions. We also presented a number of algorithms, both exact and approximate, for implementing statistical testing using this framework. We analyzed the approximate algorithm and proved that is has polynomial worst-case execution time. We also experimentally verified that its accuracy improvement is close to the exact one while providing orders of magnitude faster execution, making possible its use for causal discovery in large domains. Overall, our experimental evaluation demonstrated statistically significant improvements in the accuracy of causal discovery for the overwhelming majority of sampled, benchmark and real-world data sets.

Appendix A. Computability of the Argumentative Independence Test

In this appendix we prove that the argumentative test terminates, a property that we call its *computability*. Some of the theorems and lemmas presented are not original work but adaptations of well known properties of relations. We include them to allow a complete exposition of the proof of computability, given by Theorem 6. We first introduce some notation. We denote independence propositions (e.g., $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$) by σ and their negation (e.g., $(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})$) by $\neg\sigma$. We abbreviate their corresponding propositional arguments $(\{\sigma\}, \sigma)$ and $(\{\neg\sigma\}, \neg\sigma)$ by a_σ and $a_{\neg\sigma}$, respectively, and we will refer to $a_{\neg\sigma}$ as the *negation* of a_σ (and vice versa). Also, we use the predicates $A(a)$, $R(a)$, $Ab(a)$ to denote the fact the argument a is accepted, rejected, or in abeyance, respectively.

For completeness we repeat here the definition of strict and transitive preference relation.

Definition 11 *We say preference relation π over arguments is* strict *if the ordering of arguments induced by it is strict and total, that is, for every pair of arguments a and b,*

$$a \gg_\pi b \iff \neg\big(b \gg_\pi a\big). \tag{27}$$

Definition 12 *We say that preference relation π over arguments is* transitive *if, for every three arguments a, b and c,*

$$\big(a \gg_\pi b\big) \wedge \big(b \gg_\pi c\big) \implies \big(a \gg_\pi c\big).$$

Lemma 17 *A strict preference relation π satisfies the condition that for every pair of arguments such that a defeats b and b defeats a, it is the case that a attacks b or b attacks a, that is, at least one of a and b attacks the other.*

Proof We prove by contradiction: Let us assume that a defeats b and b defeats a but neither a attacks b nor b attacks a. By definition of the attack relation (Definition 13),

$$\neg\big(a \textit{ attacks } b\big) \implies \neg\big(\neg(b \gg_\pi a)\big) \implies b \gg_\pi a$$

and

$$\neg\big(b \textit{ attacks } a\big) \implies \neg\big(\neg(a \gg_\pi b)\big) \implies a \gg_\pi b.$$

However, this is a contradiction since, by assumption, the preference ordering is strict, and therefore it cannot be true that both $a \gg_\pi b$ and $b \gg_\pi a$ are true at the same time. ■

Lemma 18 *A strict preference π satisfies the condition that for every pair a and b of arguments, it is not the case that both a attacks b and b attacks a, that is, there can be no mutual attack.*

Proof We prove by contradiction. Let us consider two mutually attacking arguments a and b. By the definition of the attack relation, and because π is a total order, we have that

$$a \text{ attacks } b \implies \neg(b \gg_\pi a) \implies \big(a \gg_\pi b \vee a \equiv_\pi b\big)$$

and

$$b \text{ attacks } b \implies \neg(a \gg_\pi b) \implies \big(b \gg_\pi a \vee b \equiv_\pi a\big)$$

where $a \equiv_\pi b$ means a is equally preferable to b. However, equality of preference is not possible in a strict preference relation. Therefore it must be the case that $a \gg_\pi b$ and $b \gg_\pi a$, which is a contradiction of Eq. (27), again due to strictness. ■

We next prove that no argument is in abeyance if the preference relation over arguments is strict. For that, we first prove that an argument in abeyance is always attacked by at least another argument in abeyance.

Lemma 7 *For every argument a,*

$$Ab(a) \implies \exists b \in attackers(a), Ab(b).$$

Proof By definition, an argument a is in abeyance if it is neither accepted nor rejected. Applying the definitions of acceptance and rejection and manipulating the Boolean formulae we obtain,

$$\begin{aligned}
Ab(a) &\iff \neg A(a) \wedge \neg R(a) \\
&\iff \neg\big(\forall b \in attackers(a), R(b)\big) \wedge \neg\big(\exists b \in attackers(a), A(b)\big)\big) \\
&\iff \big(\exists b \in attackers(a), \neg R(b)\big) \wedge \big(\forall b \in attackers(a), \neg A(b)\big)\big) \\
&\iff \big(\exists b \in attackers(a), (A(b) \vee Ab(b))\big) \wedge \big(\forall b \in attackers(a), \neg A(b)\big)\big) \\
&\iff \big(\exists b \in attackers(a), Ab(b)\big) \wedge \big(\forall b \in attackers(a), \neg A(b)\big)\big) \\
&\implies \exists b \in attackers(a), Ab(b).
\end{aligned}$$

■

Definition 16 *An* attack sequence *is a sequence $\langle a_1, a_2, \ldots, a_n\rangle$ of n arguments such that for every $2 \leq i \leq n$, a_i attacks a_{i-1}.*

Lemma 19 *Let $\langle \mathcal{A}, \mathcal{R}, \beta\rangle$ be a PAF with a strict and transitive preference relation π. Then, no argument can appear more than once in any attack sequence, that is, for every attack sequence $\langle a_1, a_2, \ldots, a_n\rangle$ and every pair of integers $i, j \in [1, n]$ such that $i \neq j$, $a_i \neq a_j$.*

Proof
We first note that by definition of the attack relation, it must be the case that for any two consecutive arguments a_i, a_{i+1}, it is true that $\neg(a_i \gg_\pi a_{i+1})$. Since π is strict, this is equivalent to $a_{i+1} \gg_\pi a_i$ (c.f. Eq. (27)). That is,

$$a_n \gg_\pi a_{n-1} \gg_\pi \ldots \gg_\pi a_2 \gg_\pi a_1 \tag{28}$$

We now assume, for contradiction, there exists an argument $a^\star$ that appears twice in the attack sequence at indexes $i^\star$ and $j^\star$, that is,

$$\exists\, i^\star, j^\star \in [1, n], i^\star \neq j^\star, \text{ such that } a_{i^\star} = a_{j^\star} = a^\star.$$

Since no argument defeats itself, it cannot attack itself, and thus the smallest possible attack sequence with a repeated argument must have at least length 3. From this fact, Eq. (28), and transitivity, there must exist an argument $b \neq a^\star$ such that $a^\star \gg_\pi b \gg_\pi a^\star$. This last fact implies that $a^\star \gg_\pi b$ and $b \gg_\pi a^\star$ must hold, which contradicts strictness (Eq. (27)).

■

A corollary of this lemma is the following theorem.

Theorem 7 *Every attack sequence $\langle a_1, a_2, \ldots, a_n \rangle$ in a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with strict and transitive π, and finite $\mathcal{A}$ is finite.*

Proof Follows directly from Lemma 19 and the fact that $\mathcal{A}$ is finite. ■

We can now prove the main result of this section in the following theorem.

Theorem 6 *Given an arbitrary triplet $t = (X, Y \mid \mathbf{Z})$, and a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with a strict and transitive preference relation π, and finite arguments set $\mathcal{A}$, the top-down algorithm of Algorithm 3 run for input t over $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ terminates.*

Proof In the tree traversed by the top-down algorithm, any path from the root to a leaf is an attack sequence. Since for strict and transitive π, and finite $\mathcal{A}$ each such sequence is finite, the algorithm always terminates. ■

Appendix B. Validity of the Argumentative Independence Test

In this section we prove the property of the argumentative independence test of deciding that an input triplet $(X, Y \mid \mathbf{Z})$ evaluates to either independence or dependence, but not both or neither. We call this property the *validity* of the test.

We start we proving that under the assumption of a strict and transitive preference relation, no argument is in abeyance.

Theorem 5 *Let $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ be a PAF with a strict and transitive preference relation π. Then no argument $a \in \mathcal{A}$ is in abeyance.*

Proof Let us assume, for contradiction, that there is an argument a in abeyance. From Lemma 7, not only a has an attacker in abeyance, say argument b, but b also has an attacker in abeyance, and so on. That is, we can construct an attack sequence starting at a that contains only arguments in abeyance. Moreover, this sequence must be infinite,

since the lemma assures as we always have at least one attacker in abeyance. This is in direct contradiction with Theorem 7. ■

Corollary 20 *For every argument a in a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with strict and transitive π,*

$$A(a) \iff \neg R(a).$$

We now prove a number of lemmas that hold only for the sub-class of propositional arguments (arguments whose support contains only one proposition, equal to the head of that argument). We start with a lemma that demonstrates that it cannot be the case that an attacker of a propositional argument a_σ and an attacker of its negation $a_{\neg\sigma}$ do not attack each other. The former must attack the latter or vice versa.

Lemma 21 *Let $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ be a PAF with a strict preference relation π, $a_\sigma \in \mathcal{A}$ be a propositional argument, and $a_{\neg\sigma}$ its negation. For every pair of arguments b and c that attacks a_σ and $a_{\neg\sigma}$ respectively,*

$$(b \text{ attacks } c) \vee (c \text{ attacks } b).$$

Proof Since a_σ and $a_{\neg\sigma}$ are propositional arguments, their support contains the head and only the head, and thus any defeater (i.e., rebutter or undercutter) must have as head $\neg\sigma$ and σ, respectively, that is, the head of b must be $\neg\sigma$ and the head of c must be σ. Thus, b rebuts (and thus defeats) c and vice versa. The lemma then follows directly from Lemma 17. ■

Lemma 22 *Let $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ be a PAF with a strict preference relation π, and a_σ and $a_{\neg\sigma}$ be a propositional argument and its negation. Then,*

$$R(a_\sigma) \implies \neg R(a_{\neg\sigma}).$$

Proof By assumption, $R(a_\sigma)$. We assume, for contradiction, that $R(a_{\neg\sigma})$. Therefore, by the definition of rejection, $\exists b \in \mathit{attackers}(a_\sigma)$ such that $A(b)$, and $\exists c \in \mathit{attackers}(a_{\neg\sigma})$ such that $A(c)$. By Lemma 21 b *attacks* c or c *attacks* b. In either case, an accepted argument is attacking an accepted argument, which contradicts the definition of acceptance. ■

Lemma 23 *Given a PAF $\langle \mathcal{A}, \mathcal{R}, ß \rangle$ with a strict preference relation π, every propositional argument $a_\sigma \in \mathcal{A}$ satisfies*

$$A(a_\sigma) \implies \neg A(a_{\neg\sigma})$$

Proof We prove by contradiction. Let us assume that both a_σ and $a_{\neg\sigma}$ are accepted. Since a_σ and $a_{\neg\sigma}$ are propositional arguments, they defeat each other. Then, by Lemma 17 a_σ attacks $a_{\neg\sigma}$ or vice versa. In either case an accepted argument has an accepted attacker, which is a contradiction. ■

We now prove Theorem 4 that was introduced in Section 4, reproduced here for convenience.

Theorem 4 *Given a PAF $\langle \mathcal{A}, \mathcal{R}, \beta \rangle$ with a strict and transitive preference relation π, every propositional argument $a_\sigma \in \mathcal{A}$ and its negation $a_{\neg\sigma}$ satisfy*

$$A(a_\sigma) \iff R(a_{\neg\sigma}).$$

Proof The ($\implies$) direction follows from Lemma 23 and Theorem 5. The ($\impliedby$) direction follows from Lemma 22 and Theorem 5. ■

In Section 4 we defined the following semantics for deciding on the dependence or independence of an input triplet $(X, Y \mid \mathbf{Z})$:

$$\begin{aligned}(\{(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})\}, (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})) \text{ is accepted} &\iff (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \implies (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \\ (\{(X \perp\!\!\!\perp Y \mid \mathbf{Z})\}, (X \perp\!\!\!\perp Y \mid \mathbf{Z})) \text{ is accepted} &\iff (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \text{ is accepted} \implies (X \perp\!\!\!\perp Y \mid \mathbf{Z})\end{aligned} \tag{29}$$

where acceptance is defined over an independence-based PAF as defined in Section 3.3. For this argumentative test of independence to be valid, its decision must be non-ambiguous, that is, it must decide either independence or dependence, but not both or neither. For that, exactly one of the antecedents of the above implications must be true. Formally:

Theorem 3 *For any input triplet $\sigma = (X, Y \mid \mathbf{Z})$, the argumentative independence test defined by Eq. (29) produces a non-ambiguous decision, that is, it decides that σ evaluates to either independence or dependence, but not both or neither.*

Proof Let us denote $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ by $\sigma_{\mathtt{t}}$ and $(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})$ by $\sigma_{\mathtt{f}}$. Since strictness and transitivity of the independence preference relation hold (proved in Section 3.3, lemmas 14 and 15 respectively), Theorems 4 and 5 hold as well. From Theorem 5 we know that neither of the propositional arguments is in abeyance. Thus, since $a_{\sigma_{\mathtt{t}}}$ corresponds to the negation of $a_{\sigma_{\mathtt{f}}}$ it follows from Theorem 4 that exactly one of them is accepted. ■

References

H. Abdi. The Bonferonni and Šidák corrections for multiple comparisons. In Neil Salkind, editor, *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage, 2007.

A. Agresti. *Categorical Data Analysis*. Wiley, 2nd edition, 2002.

L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–215, 2002.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, 57(1):289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188, 2001.

W. G. Cochran. Some methods of strengthening the common χ^2 tests. *Biometrics*, 10: 417–451, 1954.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

C. L. Blake D. J. Newman, S. Hettich and C. J. Merz. UCI repository of machine learning databases. *Irvine, CA: University of California, Department of Information and Computer Science*, 1998.

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society*, 41:1–31, 1979.

P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence*, 77:321–357, 1995.

P. Gärdenfors. *Belief Revision*. Cambridge Computer Tracts. Cambridge University Press, Cambridge, 1992.

P. Gärdenfors and H. Rott. Belief revision. In Gabbay, D. M., Hogger, C. J. and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4. Clarendon Press, Oxford, 1995.

S. Hettich and S. D. Bay. The UCI KDD archive. *Irvine, CA: University of California, Department of Information and Computer Science*, 1999.

Y. Hochberg. A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, December 1988.

J. S. Ide, F. G. Cozman, and F. T. Ramos. Generating random Bayesian networks with constraints on induced width. *Brazilian Symposium on Artificial Intelligence, Recife, Pernambuco, Brazil*, 2002.

A. C. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9(4):515–562, 1999.

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

R. P. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence*, 2:100–106, 1987.

J. P. Martins. Belief revision. In Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, pages 110–116. John Wiley & Sons, New York, second edition, 1992.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 2nd edition, 1988.

J. Pearl and A. Paz. GRAPHOIDS: A graph-based logic for reasoning about relevance relations. Technical Report 850038 (R-53-L), Cognitive Systems Laboratory, University of California, 1985.

J. L. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

H. Prakken. *Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoning in Law.* Kluwer Law and Philosophy Library, Dordrecht, 1997.

H. Prakken and G. Vreeswijk. *Logics for Defeasible Argumentation*, volume 4 of *Handbook of Philosophical Logic*. Kluwer Academic Publishers, Dordrecht, 2 edition, 2002.

S. C. Shapiro. Belief revision and truth maintenance systems: An overview and a proposal. Technical Report CSE 98-10, Dept of Computer Science and Engineering, State University of New York at Buffalo, 1998.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning Series. MIT Press, 2nd edition, January 2000.

J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B (Methodological)*, 64(3):479–498, 2002.

M. Studený. Conditional independence relations have no finite complete characterization. In *Transactions of the 11th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, volume B, pages 377–396, 1991.

F. Toni and A. C. Kakas. Computing the acceptability semantics. In A. Nerode, editor, *3rd International Conference on Logic Programming and Non-monotonic Reasoning*, volume 928 of *Lecture Notes in Artificial Intelligence*, pages 401–415. Springer Verlag, 1995.

JMLR: Workshop and Conference Proceedings 10:699–718, 2009 Submitted 6/07; Revised 2/08; Published 03/09

Properties of Monotonic Effects on Directed Acyclic Graphs

Tyler J. VanderWeele
University of Chicago
Chicago, IL 60615, USA

James M. Robins
Harvard School of Public Health
Boston, MA 02115, USA

Editor: Constantin Aliferis

Abstract

Various relationships are shown hold between monotonic effects and weak monotonic effects and the monotonicity of certain conditional expectations. Counterexamples are provided to show that the results do not hold under less restrictive conditions. Monotonic effects are furthermore used to relate signed edges on a causal directed acyclic graph to qualitative effect modification. The theory is applied to an example concerning the direct effect of smoking on cardiovascular disease controlling for hypercholesterolemia. Monotonicity assumptions are used to construct a test for whether there is a variable that confounds the relationship between the mediator, hypercholesterolemia, and the outcome, cardiovascular disease.

Keywords: Bayesian networks, conditional expectation, covariance, directed acyclic graphs, effect modification, monotonicity

1. Introduction

Several papers have considered various monotonicity relationships on Bayesian networks or directed acyclic graphs. Wellman (1990) introduced the notion of qualitative causal influence and derived various resulting concerning the propagation of qualitative influences, the preservation of monotonicity under edge reversal, the necessity of first order stochastic dominance for propagating influences and the propagation of sub-additive and super-additive relationships on probabilistic networks. Druzdzel and Henrion (1993) developed a polynomial time algorithm for reasoning in qualitative probabilistic network, based on local sign propagation. More recently, van der Gaag et al. (2004) showed that identifying whether a network exhibits various monotonicity properties is coNP^{PP}- complete. VanderWeele and Robins (2009) introduced the concept of a monotonic effect which is closely related to Wellman's qualitative influence and considered the relationship between monotonicity properties and causal effects, covariance, bias and confounding. In this paper we develop a number of probabilistic properties concerning monotonic effects and weak monotonic effects. Some of these properties give rise to certain inequality constraints that could be used to test for the presence of hidden or unmeasured confounding variables. These inequality constraints which arise from monotonicity relationships provide constraints beyond those already

available in the literature (Kang and Tian, 2006). The paper is organized as follows. In Section 2 we describe the notation we will use in this paper and review the definitions concerning directed acyclic graphs. In Section 3 we present a motivating example for the theory that will be developed. In Section 4, we define the concepts of a monotonic effect and a weak monotonic effect in the directed acyclic graph causal framework, the latter essentially being equivalent to Wellman's (1990) qualitative influence. In Section 5, we give a number of results relating weak monotonic effects to the monotonicity in the conditioning argument of certain conditional expectations; we also return to the motivating example and show how the theory developed can be applied to this example. Finally, in Section 6, we give a number of results that relate weak monotonic effects to the existence of qualitative effect modifiers. Section 7 closes with some concluding remarks.

2. Notation and Directed Acyclic Graphs

Following Pearl (1995), a causal directed acyclic graph is a set of nodes $(X_1, \ldots, X_n)$ and directed edges amongst nodes such that the graph has no cycles and such that for each node X_i on the graph the corresponding variable is given by its non-parametric structural equation $X_i = f_i(pa_i, \epsilon_i)$ where pa_i are the parents of X_i on the graph and the ϵ_i are mutually independent. We will use Ω to denote the sample space for ϵ and ω to denote a particular point in the sample space. These non-parametric structural equations can be seen as a generalization of the path analysis and linear structural equation models (Pearl, 1995, 2000) developed by Wright (1921) in the genetics literature and Haavelmo (1943) in the econometrics literature. Directed acyclic graphs can be interpreted as representing causal relationships. The non-parametric structural equations encode counterfactual relationships amongst the variables represented on the graph. The equations themselves represent one-step ahead counterfactuals with other counterfactuals given by recursive substitution. The requirement that the ϵ_i be mutually independent is essentially a requirement that there is no variable absent from the graph which, if included on the graph, would be a parent of two or more variables (Pearl, 1995, 2000). Further discussion of the causal interpretation of directed acyclic graphs can be found elsewhere (Pearl, 1995, 2000; Spirtes et al., 2000; Dawid, 2002; Robins, 2003).

A path is a sequence of nodes connected by edges regardless of arrowhead direction; a directed path is a path which follows the edges in the direction indicated by the graph's arrows. A node C is said to be a common cause of A and Y if there exists a directed path from C to Y not through A and a directed path from C to A not through Y. We will say that $V_1, \ldots, V_n$ constitutes an ordered list if $i < j$ implies that V_i is not a descendant of V_j. A collider is a particular node on a path such that both the preceding and subsequent nodes on the path have directed edges going into that node i.e. both the edge to and the edge from that node have arrowheads into the node. A path between A and B is said to be blocked given some set of variables Z if either there is a variable in Z on the path that is not a collider or if there is a collider on the path such that neither the collider itself nor any of its descendants are in Z. If all paths between A and B are blocked given Z then A and B are said to be d-separated given Z. It has been shown that if A and B are d-separated given Z then A and B are conditionally independent given Z (Verma and Pearl, 1988; Geiger et al., 1990; Lauritzen et al., 1990). We will use the notation $A \coprod B|Z$ to denote that A is conditionally independent of B given Z; we will use the notation $(A \coprod B|Z)_G$ to denote that A and B are d-separated given Z on graph G. The directed acyclic graph causal framework has proven to be particularly

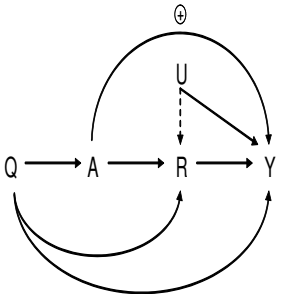


Figure 1: Motivating example concerning the estimation of controlled direct effects.

useful in determining whether conditioning on a given set of variables, or none at all, is sufficient to control for confounding. The most important result in this regard is the back-door path criterion (Pearl, 1995). A back-door path from some node A to another node Y is a path which begins with a directed edge into A. Pearl (1995) showed that for intervention variable A and outcome Y, if a set of variables Z is such that no variable in Z is a descendant of A and such that Z blocks all back-door paths from A to Y then conditioning on Z suffices to control for confounding for the estimation of the causal effect of A on Y. The counterfactual value of Y intervening to set $A = a$ we denote by $Y_{A=a}$.

3. Motivating Example

To motivate the theory we develop in this paper consider the following example.

Example 1 Suppose that Figure 1 represents a causal directed acyclic graph.

Let A denote smoking; let R hypercholesterolemia; and let Y denote cardiovascular disease. High cholesterol can lead to the narrowing of the arteries resulting in cardiovascular disease; smoking can lead to blood clots through platelet aggregation resulting in cardiovascular disease. Let Q denote some variable that confounds the relationships between smoking and cardiovascular disease and between hypercholesterolemia and cardiovascular disease (e.g. stress). Let U be some unmeasured variable which might confound the relationship between hypercholesterolemia and cardiovascular disease. The researcher is unsure whether the variable U is a cause of R and we therefore represent the edge from U to R as a dashed line. The results of Pearl (2001) imply that it is possible to estimate controlled direct effects of the form $Y_{A=a_1,R=r} - Y_{A=a_0,R=r}$ (i.e. the direct effect of smoking on cardiovascular disease controlling for hypercholesterolemia) on the graph in Figure 1 if that U is not a cause of R. Suppose that although the researcher is unsure about the presence an edge from U to R, it is known that the relationship between A and Y is monotonic in the sense that $P(Y > y|A = a, R = r, Q = q, U = u)$ is non-decreasing in a for all y, r, q and u. In Section 5, we will present theory that will allow us to derive a statistical test for the null hypothesis that there is no unmeasured variable U confounding the relationship between R and Y.

4. On the Definition of a Monotonic Effect

The definition of a monotonic effect is given in terms of a directed acyclic graph's nonparametric structural equations.

Definition 1 *The non-parametric structural equation for some node Y on a causal directed acyclic graph with parent A can be expressed as $Y = f(\widetilde{pa}_Y, A, \epsilon_Y)$ where $\widetilde{pa}_Y$ are the parents of Y other than A; A is said to have a positive monotonic effect on Y if for all $\widetilde{pa}_Y$ and ϵ_Y, $f(\widetilde{pa}_Y, A_1, \epsilon_Y) \geq f(\widetilde{pa}_Y, A_2, \epsilon_Y)$ whenever $A_1 \geq A_2$. Similarly A is said to have a negative monotonic effect on Y if for all $\widetilde{pa}_Y$ and ϵ_Y, $f(\widetilde{pa}_Y, A_1, \epsilon_Y) \leq f(\widetilde{pa}_Y, A_2, \epsilon_Y)$ whenever $A_1 \geq A_2$.*

As we have defined it above, a causal direct acyclic graph corresponds to a set of non-parametric structural equations and as such the definition of a monotonic effect given above is relative to a particular set of non-parametric structural equations. The presence of a monotonic effect is closely related to the monotonicity of counterfactual variables as is made clear by the following proposition. All proofs of all propositions and theorems are given in Appendix A.

Proposition 2 *The variable A has a positive monotonic effect on Y if and only if for all ω and all values of $\widetilde{pa}_Y$, $Y_{a_1,\widetilde{pa}_Y}(\omega) \geq Y_{a_0,\widetilde{pa}_Y}(\omega)$ whenever $a_1 \geq a_0$.*

We note that several sets of non-parametric structural equations may yield identical distributions of $X = (X_1, \ldots, X_n)$ and $\{X_{V=v}\}_{V \subseteq X, v \in supp(V)}$ (Pearl, 2000). In the context of characterizations of causal directed acyclic graphs that make reference to counterfactuals but not to non-parametric structural equations (Robins, 2003), a positive monotonic effect could instead be defined to be present if for all $\widetilde{pa}_Y$ and $a_1 \geq a_0$, $P(Y_{a_1,\widetilde{pa}_Y} \geq Y_{a_0,\widetilde{pa}_Y}) = 1$. If this latter condition holds with respect to one set of non-parametric structural equations it will hold for any set of non-parametric structural equations which yields the same distribution for X and $\{X_{V=v}\}_{V \subseteq X, v \in supp(V)}$. We note that if for $a_1 \geq a_0$ the set $\{\omega : Y_{a_1,\widetilde{pa}_Y}(\omega) < Y_{a_0,\widetilde{pa}_Y}(\omega)\}$ is of measure zero then $Y_{a_1,\widetilde{pa}_Y}$ and $Y_{a_0,\widetilde{pa}_Y}$ could be re-defined on this set so that $Y_{a_1,\widetilde{pa}_Y}(\omega) \geq Y_{a_0,\widetilde{pa}_Y}(\omega)$ for all ω and so that the distributions of X and $\{X_{V=v}\}_{V \subseteq X, v \in supp(V)}$ remain unchanged.

Because for any value ω we observe the outcome only under one particular value of the intervention variable, the presence of a monotonic effect is not identifiable. The results presented in this paper are in fact true under slightly weaker conditions which are identifiable when data on all of the directed acyclic graph's variables are observed. We thus introduce the concept of a weak monotonic effect which is a special case of Wellman's positive qualitative influence (Wellman, 1990). The definition of a weak monotonic effect does not make reference to counterfactuals and thus can be used in characterizations of causal directed acyclic graphs that do not employ the concept of counterfactuals (Spirtes et al., 2000; Dawid, 2002). The stronger notion of a monotonic effect given above is useful in the context of testing for synergistic relationships (VanderWeele and Robins, 2008).

Definition 3 *Suppose that variable A is a parent of some variable Y and let $\widetilde{pa}_Y$ denote the parents of Y other than A. We say that A has a weak positive monotonic effect on Y if the survivor function $S(y|a, \widetilde{pa}_Y) = P(Y > y|A = a, \widetilde{pa}_Y)$ is such that whenever $a_1 \geq a_0$ we have $S(y|a_1, \widetilde{pa}_Y) \geq S(y|a_0, \widetilde{pa}_Y)$ for all y and all $\widetilde{pa}_Y$; the variable A is said to have a weak negative monotonic effect on Y if whenever $a_1 \geq a_0$ we have $S(y|a_1, \widetilde{pa}_Y) \leq S(y|a_0, \widetilde{pa}_Y)$ for all y and all $\widetilde{pa}_Y$.*

Proposition 4 *If A has a positive monotonic effect on Y then A has a weak positive monotonic effect on Y.*

We note that for parent A and child Y, the definition of a weak monotonic effect coincides with Wellman's (1990) definition of positive qualitative influence when the "context" for qualitative influence is chosen to be the parents of Y other than A.

A monotonic effect is a relation between two nodes on a directed acyclic graph and as such it is associated with an edge. The definition of the sign of an edge can be given either in terms of monotonic effects or weak monotonic effects. We can define the sign of an edge as the sign of the monotonic effect or weak monotonic effect to which the edge corresponds; this in turn gives rise to a natural definition for the sign of a path.

Definition 5 *An edge on a causal directed acyclic graph from X to Y is said to be of positive sign if X has a positive monotonic effect on Y. An edge from X to Y is said to be of negative sign if X has a negative monotonic effect on Y. If X has neither a positive monotonic effect nor a negative monotonic effect on Y, then the edge from X to Y is said to be without a sign.*

Definition 6 *The sign of a path on a causal directed acyclic graph is the product of the signs of the edges that constitute that path. If one of the edges on a path is without a sign then the sign of the path is said to be undefined.*

We will call a causal directed acyclic graph with signs on those edges which allow them a signed causal directed acyclic graph. The theorems in this paper are given in terms of signed paths so as to be applicable to both monotonic effects and weak monotonic effects. One further definition will be useful in the development of the theory below.

Definition 7 *Two variables X and Y are said to be positively monotonically associated if all directed paths from X to Y or from Y to X are of positive sign and all common causes C_i of X and Y are such that all directed paths from C_i to X are of the same sign as all directed paths from C_i to Y; the variables X and Y are said to be negatively monotonically associated if all directed paths between X and Y are of negative sign and all common causes C_i of X and Y are such that all directed paths from C_i to X are of the opposite sign as all directed paths from C_i to Y.*

It has been shown elsewhere (VanderWeele and Robins, 2009) that if X and Y are positively monotonically associated then $Cov(X, Y) \geq 0$ and if X and Y are negatively monotonically associated then $Cov(X, Y) \leq 0$. We now develop several results concerning the monotonicity in the conditioning argument of certain conditional expectations.

5. Monotonic Effects and Conditional Expectations

Lemma 8 below can be proved by integration by parts and will be used in the proofs of the subsequent propositions. We will assume throughout the remainder of this paper that the random variables under consideration satisfy regularity conditions that allow for the integration by parts required in the proof of Lemma 8. If conditional cumulative distribution functions are continuously differentiable then the regularity conditions will be satisfied; the regularity conditions will also be satisfied if all variables are discrete. Härdle et al. (1998, p72) also gives relatively weak conditions under which such integration by parts is possible. Alternatively, the existence of the Lebesgue-Stieltjes integrals found in the proof of Lemma 8 suffices to allow integration by parts.

Note that Lemma 8 will always be applied either to the function $h(y,a,r) = y$ or to conditional survivor functions which will satisfy the relevant regularity conditions; thus the conditions which are required for integration by parts are only regularity conditions on the distribution of the random variables.

Lemma 8 *If $h(y,a,r)$ is non-decreasing in y and in a and $S(y|a,r) = P(Y > y|A = a, R = r)$ is non-decreasing in a for all y then $E[h(Y,A,R)|A = a, R = r]$ is non-decreasing in a.*

Proposition 9 immediately follows from Lemma 8.

Proposition 9 *Suppose that the $A \rightarrow Y$ edge, if it exists, is positive. Let X denote some set of non-descendants of Y that includes $\widetilde{pa}_Y$, the parents of Y other than A, then $E[Y|X = x, A = a]$ is non-decreasing in a for all values of x.*

Proposition 12 gives the basic result for the monotonicity of conditional expectations. For the conditional expectation of some variable Y to be monotonic in a conditioning argument A, it requires that the conditioning set includes variables that block all backdoor paths from A to Y. In order to prove Proposition 12 we will make use of the following two lemmas.

Lemma 10 *Suppose that A is a non-descendant of Y and let Q denote the set of ancestors of A or Y which are not descendants of A. Let $R = (R_1, \ldots, R_m)$ denote an ordered list of some set of nodes on directed paths from A to Y such that for each i the backdoor paths from R_i to Y are blocked by $R_1, \ldots, R_{i-1}, A$, and Q. Let $V_0 = A$ and $V_n = Y$ and let $V_1, \ldots, V_{n-1}$ be an ordered list of all the nodes which are not in R but which are on directed paths from A to Y such that at least one of the directed paths from each node to Y is not blocked by R. Let $\overline{V}_k = \{V_1, \ldots, V_k\}$ then $S(v_k|a, \overline{v}_{k-1}, q, r) = S(v_k|pa_{v_k})$.*

Lemma 11 *If under the conditions of Lemma 10 all directed paths from A to Y are positive except possibly through R then $S(y|a,q,r)$ is non-decreasing in a.*

These two lemmas allow us to prove Proposition 12 given below.

Proposition 12 *Suppose that A is a non-descendant of Y and let X denote some set of non-descendants of A that blocks all backdoor paths from A to Y. Let $R = (R_1, \ldots, R_m)$ denote an ordered list of some set of nodes on directed paths from A to Y such that for each i the backdoor paths from R_i to Y are blocked by $R_1, \ldots, R_{i-1}, A$ and X. If all directed paths from A to Y are positive except possibly through R then $S(y|a,x,r)$ and $E[y|a,x,r]$ are non-decreasing in a.*

If $R = \varnothing$ the statement of Proposition 12 is considerably simplified and is stated in the following corollary.

Corollary 13 *Let X denote some set of non-descendants of A that blocks all backdoor paths from A to Y. If all directed paths between A and Y are positive then $S(y|a,x)$ and $E[y|a,x]$ are non-decreasing in a.*

Lemma 11 and Proposition 12 are generalizations of results given by Wellman (1990) and Druzdzel and Henrion (1993). In particular, in Lemma 11 if $R = \varnothing$, then the result follows immediately from repeated application of Theorems 4.2 and 4.3 in Wellman (1990) or more directly from the work of Druzdzel and Henrion (1993, Theorem 4). Lemma 11 generalizes the results of Wellman (1990) and Druzdzel and Henrion (1993) by allowing for conditioning on nodes $R = (R_1, \ldots, R_m)$ which are on directed paths

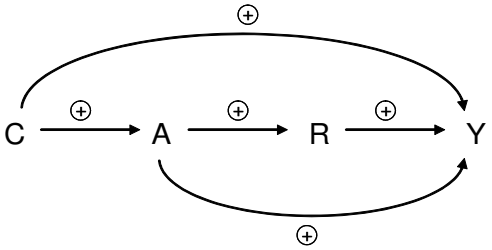


Figure 2: Example illustrating Propositions 12–15.

from A to Y. Proposition 12 further generalizes Lemma 11 by replacing the set Q in Lemma 11 which consists of the set of ancestors of A or Y which are not descendants of A with some other set X which consists of some set of non-descendants of A that blocks all backdoor paths from A to Y.

Propositions 14–18 relax the condition that the conditioning set includes variables that block all backdoor paths A to Y and impose certain other conditions; the proofs of each of these propositions make use of Proposition 12.

Proposition 14 *Suppose that A is not a descendant of Y, that A is binary, and that A and Y are positively monotonically associated then $E[Y|A]$ is non-decreasing in A.*

Proposition 15 *Suppose that A is not a descendant of Y, that Y is binary, and that A and Y are positively monotonically associated then $E[A|Y]$ is non-decreasing in Y.*

Propositions 14 and 15 require that conditioning variable be binary. Counterexamples can be constructed to show that if the conditioning variable is not binary then the conditional expectation may not be non-decreasing in the conditioning argument even if A and Y are positively monotonically associated (see Appendix B, Counterexamples 1 and 2).

Propositions 14 and 15 can be combined to give the following corollary which makes no reference to the ordering of A and Y.

Corollary 16 *Suppose that A is binary and that A and Y are positively monotonically associated then $E[Y|A]$ is non-decreasing in A.*

Example 2 Consider the signed directed acyclic graph given in Figure 2.

By Proposition 12, we have that $E[Y|A = a, C = c, R = r]$ and $E[Y|A = a, C = c]$ are non-decreasing in a. If A is binary then by Proposition 14, it is also the case that $E[Y|A = a]$ is non-decreasing in a. If Y is binary, then by Proposition 15, $E[A|Y = y]$ is non-decreasing in y. The monotonicity of $E[Y|A = a, C = c, R = r]$ and $E[Y|A = a, C = c]$ also follow directly from the results of Wellman (1990) and Druzdzel and Henrion (1993); the monotonicity of $E[Y|A = a]$ and $E[A|Y = y]$ do not.

Propositions 17 and 18 consider the monotonicity of conditional expectations while conditioning on variables other than the variable in which monotonicity holds but not conditioning on variables that are sufficient to block all backdoor paths between A and Y. Propositions 17 and 18 generalize Propositions 14 and 15 respectively.

Proposition 17 *Suppose that A is not a descendant of Y and that A is binary. Let Q be some set of variables that are not descendants of Y nor of A and let C be the common causes of A and*

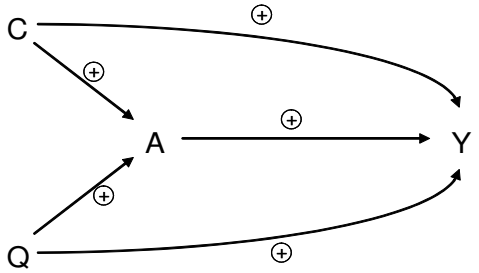


Figure 3: Example illustrating Propositions 17 and 18.

Y not in Q. If all directed paths from A to Y are of positive sign and all directed paths from C to A not through Q are of the same sign as all directed paths from C to Y not through $\{Q, A\}$ *then* $E[Y|A, Q]$ *is non-decreasing in A.*

Proposition 18 is similar to Proposition 17 but the conditional expectation $E[A|Y, Q]$ is considered rather than $E[Y|A, Q]$ and Y rather than A is assumed to be binary. The form of the proof differs.

Proposition 18 *Suppose that A is not a descendant of Y and that Y is binary. Let Q be some set of variables that are not descendants of Y nor of A and let C be the common causes of A and Y not in Q. If all directed paths from A to Y are of positive sign and all directed paths from C to A not through Q are of the same sign as all directed paths from C to Y not through* $\{Q, A\}$ *then* $E[A|Y, Q]$ *is non-decreasing in Y.*

Propositions 17 and 18 can be combined to give the following corollary which makes no reference to the ordering of A and Y.

Corollary 19 *Suppose that A is binary. Let Q be some set of variables that are not descendants of Y nor of A and let C be the common causes of A and Y not in Q. If all directed paths from A to Y (or from A to Y) are of positive sign and all directed paths from C to A not through* $\{Q, Y\}$ *are of the same sign as all directed paths from C to Y not through* $\{Q, A\}$ *then* $E[Y|A, Q]$ *is non-decreasing in Y.*

Example 3 Consider the signed directed acyclic graph given in Figure 3.

If A is binary, then by Proposition 17, $E[Y|A = a, C = c, Q = q]$, $E[Y|A = a, Q = q]$, $E[Y|A = a, C = c]$ and $E[Y|A = a]$ are all non-decreasing in a. If Y is binary then by Proposition 18, $E[A|Y = y, C = c, Q = q]$, $E[A|Y = y, Q = q]$, $E[A|Y = y, C = c]$ and $E[A|Y = y]$ are all non-decreasing in y. The monotonicity of $E[Y|A = a, C = c, Q = q]$ follows directly from the results of Wellman (1990) and Druzdzel and Henrion (1993); the monotonicity of the other conditional expectations do not.

We now return to Example 1 concerning potential unmeasured confounding in the estimation of controlled direct effects.

Example 1 (Revisited). Consider once again the causal directed acyclic graph given in Figure 1. Suppose that we may assume that A has a weak monotonic effect on Y. Under the null hypothesis that U is not a cause of R (i.e. does not confound the relationship between R and Y) we could conclude by Proposition 12 that $E[Y|A = a, R = r, Q = q]$ is non-decreasing in a for all r and q. Under the alternative hypothesis

that U is a cause of R, we could not apply Proposition 12 because of the unblocked backdoor path $R - U - Y$ between R and Y. The monotonicity relationship would thus not necessarily hold. Consequently, if $E[Y|A = a, R = r, Q = q]$ were found not to be monotonic in a then we could reject the null hypothesis that U is not a cause of R. Note that the monotonicity of $E[Y|A = a, R = r, Q = q]$ in a also follows from the results of Wellman (1990) and Druzdzel and Henrion (1993). If, however, there were an edge from U to Q for example, or in more complicated scenarios, the results of Wellman (1990) and Druzdzel and Henrion (1993) would no longer suffice to conclude the monotonicity of $E[Y|A = a, R = r, Q = q]$ in a; one would need to employ Proposition 12.

We now construct a simple statistical test in the case that A, R and Y are all binary (cf. Robins and Greenland, 1992) of the null hypothesis that U is absent from Figure 1. Let n_{ijq} denote the number of individuals in stratum $Q = q$ with $A = i$ and $R = j$ and let let d_{ijq} denote the number of individuals in stratum $Q = q$ with $A = i$ and $R = j$ and $Y = 1$. Let p_{ijq} denote the true probability $P(Y = 1|A = i, R = j, Q = q)$. From the null hypothesis that U is absent from Figure 1, it follows by Proposition 12 that $p_{1jq} - p_{0jq} \leq 0$ for all j and q. Thus we have $d_{ijq} \sim Bin(n_{ijq}, p_{ijq})$ with $\mathbb{E}[\frac{d_{ijq}}{n_{ijq}}] = p_{ijq}$ and $Var(\frac{d_{ijq}}{n_{ijq}}) = \frac{p_{ijq}(1-p_{ijq})}{n_{ijq}}$. By the central limit central limit theorem $\frac{(\frac{d_{1jq}}{n_{1jq}} - \frac{d_{0jq}}{n_{0jq}}) - (p_{1jq} - p_{0jq})}{\sqrt{\frac{p_{1jq}(1-p_{1jq})}{n_{1jq}} + \frac{p_{0jq}(1-p_{0jq})}{n_{0jq}}}} \dot{\sim} N(0,1)$ and by Slutsky's theorem we have $\frac{(\frac{d_{1jq}}{n_{1jq}} - \frac{d_{0jq}}{n_{0jq}}) - (p_{1jq} - p_{0jq})}{\sqrt{\frac{d_{1jq}(n_{1jq} - d_{1jq})}{n_{1jq}^3} + \frac{d_{0jq}(n_{0jq} - d_{0jq})}{n_{0jq}^3}}} \dot{\sim} N(0,1)$. To test the null hypothesis that the edge from U to R is absent from Figure 1 one may thus use the test statistic $\frac{(\frac{d_{1jq}}{n_{1jq}} - \frac{d_{0jq}}{n_{0jq}})}{\sqrt{\frac{d_{1jq}(n_{1jq} - d_{1jq})}{n_{1jq}^3} + \frac{d_{0jq}(n_{0jq} - d_{0jq})}{n_{0jq}^3}}}$ with critical regions of the form: $\{\frac{(\frac{d_{1jq}}{n_{1jq}} - \frac{d_{0jq}}{n_{0jq}})}{\sqrt{\frac{d_{1jq}(n_{1jq} - d_{1jq})}{n_{1jq}^3} + \frac{d_{0jq}(n_{0jq} - d_{0jq})}{n_{0jq}^3}}} > Z_{1-\alpha}\}$ to carry out a one-sided (upper tail) test. The derivation of the power of such a test would require providing explicit structural equations for each of the variables in the model. Similar tests could be constructed for other scenarios. We note that if the test fails to reject the null, one cannot conclude that the arrow from U to R is absent; if the inequality $E[Y|A = a_1, R = r, Q = q] \leq E[Y|A = a_2, R = r, Q = q]$ holds for all $a_1 \leq a_2$ this is potentially consistent with both the presence and the absence of an edge from U to R. If, however, the test rejects the null then one can conclude that an edge from U to R must be present, provided the other model assumptions hold. With observational data, the assumption that no unmeasured confounding variable is present can be falsified but it cannot be verified regardless of the approach one takes. It is nevertheless worthwhile testing any empirical implications of the no unmeasured confounding variables assumptions which can be derived, such as those following from Proposition 12.

Tian and Pearl (2002) and Kang and Tian (2007) derived various equality constraints that arise from causal directed acyclic graphs with hidden variables; Kang and Tian (2006) derived various inequality constraints that arise from causal directed acyclic graphs with hidden variables. We note that the inequality constraint $E[Y|A = a_1, R = r, Q = q] \leq E[Y|A = a_2, R = r, Q = q]$ for $a_1 \leq a_2$ does not follow from the results of Tian and Pearl (2002) or Kang and Tian (2006, 2007). The equality and inequality

constraints which follow from their work will apply to all causal models consistent with the directed acyclic graph in Figure 1 (without the sign); the inequality constraint $E[Y|A = a_1, R = r, Q = q] \leq E[Y|A = a_2, R = r, Q = q]$ follows only if it can be assumed in Figure 1 that A has a weak positive monotonic effect on Y. More generally, the results in this paper do not provide an alternative set of constraints but rather a supplementary set of constraints to those of Tian and Pearl (2002) and Kang and Tian (2006, 2007).

6. Effect Modification and Monotonic Effects

If when conditioning on a particular variable, the sign of the effect of another variable on the outcome varies between strata of the conditioning variable, then the conditioning variable is said to be a qualitative effect modifier. The following definition gives the condition for qualitative effect modification more formally.

Definition 20 *A variable Q is said to be an effect modifier for the causal effect of A on Y if Q is not a descendant of A and if there exist two levels of A, a_0 and a_1 say, such that $E[Y_{A=a_1}|Q = q] - E[Y_{A=a_0}|Q = q]$ is not constant in q. Furthermore Q is said to be a qualitative effect modifier if there exist two levels of A, a_0 and a_1, and two levels of Q, q_0 and q_1, such that $sign(E[Y_{A=a_1}|Q = q_1] - E[Y_{A=a_0}|Q = q_1]) \neq sign(E[Y_{A=a_1}|Q = q_0] - E[Y_{A=a_0}|Q = q_0])$.*

Monotonic effects and weak monotonic effects are closely related to the concept of qualitative effect modification. Essentially, the presence of a monotonic effect precludes the possibility of qualitative effect modification. This is stated precisely in Propositions 21 and 23.

Proposition 21 *Suppose that some parent A_1 of Y is such that the $A_1 - Y$ edge is of positive sign then there can be no other parent, A_2, of Y which is a qualitative effect modifier for causal effect of A_1 on Y, either unconditionally or within some stratum $C = c$ of the parents of Y other than A_1 and A_2.*

A similar result clearly holds if the $A_1 - Y$ edge is of negative sign. We give the contrapositive of Proposition 21 as a corollary.

Corollary 22 *Suppose that some parent of Y, A_2, is a qualitative effect modifier for causal effect of another parent of Y, A_1, either unconditionally or within some stratum $C = c$ of the parents of Y other than A_1 and A_2 then A_1 can have neither a weak positive monotonic effect nor a weak negative monotonic effect on Y.*

If there are intermediate variables between A and Y then Proposition 21 can be generalized to give Proposition 23.

Proposition 23 *Suppose that all directed paths from A to Y are of positive sign (or are all of negative sign) then there exists no qualitative effect modifier Q on the directed acyclic graph for the causal effect of A on Y.*

Example 4 Consider the signed directed acyclic graph given in Figure 4 in which the $A - Y$ edge is of positive sign.

It can be shown that any of Q_1, Q_2, Q_3, Q_4 or Q_5 can serve as effect modifiers for the causal effect of A on Y (VanderWeele and Robins, 2007). However, by Proposition 21

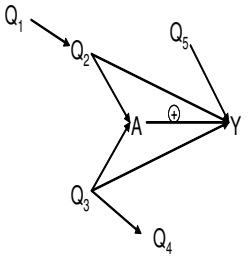


Figure 4: Example illustrating the use of Propositions 21 and 23.

or 23, since A has a (weak) monotonic effect on Y, none of Q_1, Q_2, Q_3, Q_4 or Q_5 can serve as *qualitative* effect modifiers for the causal effect of A on Y. Conversely, if it is found that one of Q_1, Q_2, Q_3, Q_4 or Q_5 *is* a qualitative effect modifier for the causal effect of A on Y then the $A - Y$ edge cannot be of positive (or negative) sign.

7. Concluding Remarks

In this paper we have related weak monotonic effects to the monotonicity of certain conditional expectations in the conditioning argument and to qualitative effect modification. When the variables on a causal directed acyclic graph exhibit weak monotonic effects the results can be used to construct tests for the presence of unmeasured confounding variables. Future work could examine whether it is possible to weaken the restrictions on R in Proposition 12; another area of future research would include developing an algorithm for what relationships need systematic evaluation in order to test for particular confounding patterns; further research could also be done on the derivation of statistical tests of the type considered at the end of Section 5 for cases in which A, R and Y are not binary and for dealing with issues related to multiple testing problems.

Acknowledgments

We would like to thank the editors and three anonymous referees for helpful comments on this paper.

Appendix A. Proofs.

Proof of Proposition 2.

By the definition of a non-parametric structural equation, $Y_{a,\widetilde{pa}_Y}(\omega) = f(\widetilde{pa}_Y, a, \epsilon_Y(\omega))$ and from this the result follows.

Proof of Proposition 4.

Since A has a positive monotonic effect on Y, for any $a_1 \geq a_0$ we have that $S(y|a_1, \widetilde{pa}_Y) = P(Y > y|a_1, \widetilde{pa}_Y) = P\{f(\widetilde{pa}_Y, a_1, \epsilon_Y) > y\} \geq P\{f(\widetilde{pa}_Y, a_0, \epsilon_Y) > y\} = P(Y > y|a_0, \widetilde{pa}_Y) = S(y|a_1, \widetilde{pa}_Y)$.

Proof of Lemma 8.

For $a \geq a'$ we have

$$
\begin{aligned}
&E[h(Y,A,R)|A=a,R=r] - E[h(Y,A,R)|A=a',R=r] \\
&\qquad = \int_{y=-\infty}^{y=\infty} h(y,a,r)dF(y|a,r) - \int_{y=-\infty}^{y=\infty} h(y,a',r)dF(y|a',r) \\
&= \int_{y=-\infty}^{y=\infty} h(y,a,r)d\{F(y|a,r) - F(y|a',r)\} + \int_{y=-\infty}^{y=\infty} \{h(y,a,r) - h(y,a',r)\}dF(y|a',r) \\
&= [h(y,a,r)\{F(y|a,r) - F(y|a',r)\}]_{y=-\infty}^{y=\infty} - \int_{y=-\infty}^{y=\infty} \{F(y|a,r) - F(y|a',r)\}dh(y,a,r) \\
&\qquad + \int_{y=-\infty}^{y=\infty} \{h(y,a,r) - h(y,a',r)\}dF(y|a',r) \\
&= \int_{y=-\infty}^{y=\infty} \{S(y|a,r) - S(y|a',r)\}dh(y,a,r) + \int_{y=-\infty}^{y=\infty} \{h(y,a,r) - h(y,a',r)\}dF(y|a',r).
\end{aligned}
$$

This final expression is non-negative since the integrands of both integrals are non-negative for $a \geq a'$.

Proof of Proposition 9.

We have that $E[Y|X=x,A=a] = E[Y|\widetilde{pa}_Y, A=a]$ and since A has a (weak) positive monotonic effect on Y, we have that $S(y|a,\widetilde{pa}_Y)$ is non-decreasing in a and it follows from Lemma 8 that $E[Y|X=x,A=a] = E[Y|\widetilde{pa}_Y, A=a]$ is non-decreasing in a.

Proof of Lemma 10.

We will say a path from A to B is a frontdoor path from A to B if the path begins with a directed edge with the arrowhead pointing out of A. Let Q^k and R^k be the subsets of Q and R respectively that are ancestors of $\overline{V}_k$. We will show that

$$
\begin{aligned}
S(v_k|a,v_1,\ldots,v_{k-1},q,r) &= S(v_k|a,v_1,\ldots,v_{k-1},q,r^k) \\
&= S(v_k|a,v_1,\ldots,v_{k-1},q^k,r^k) = S(v_k|pa_{v_k}).
\end{aligned}
$$

If $R^k = R$, the first equality holds trivially. Suppose that $R^k \neq R$ so that R_m is not an ancestor of V_k. All frontdoor paths from R_m to V_k must include a collider since R_m is not an ancestor of V_k. This collider will not be in $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$ since all these variables are non-descendants of R_m. Thus all frontdoor paths from R_m to V_k will be blocked given $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$. All backdoor paths from R_m to V_k with an edge going into V_k will be blocked given $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$ by pa_{V_k}; note by hypothesis it can be seen that pa_{V_k} will be contained by the variables $A, V_1, \ldots, V_{k-1}, Q, R^k$ since there is a directed path from V_k to Y and Q includes all ancestors of Y not on directed paths from A to Y. All backdoor paths from R_m to V_k with an edge going out from V_k will be blocked given $A, Q, R_1, \ldots, R_{m-1}$ by hypothesis; otherwise there would be a backdoor path from R_m through V_k to Y not blocked by $A, Q, R_1, \ldots, R_{m-1}$. But all backdoor paths from R_m to V_k with an edge going out from V_k which are blocked by $A, Q, R_1, \ldots, R_{m-1}$ will also be blocked by $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$. This is because such a path concluding with an edge going out from V_k which is blocked by $A, Q, R_1, \ldots, R_{m-1}$ but not blocked by $A, V_1, \ldots,$

$V_{k-1}, Q, R_1, \ldots, R_{m-1}$ would require that one of $V_1, \ldots, V_{k-1}$, say V_p, be a collider on the path or a descendant of a collider. If one of $V_1, \ldots, V_{k-1}$ were a collider then the path would in fact be blocked by the parents of the collider since all the parents of $V_1, \ldots, V_{k-1}$ are in $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$. If one of $V_1, \ldots, V_{k-1}$, say V_p, were a descendant of the collider then none of the directed paths from the collider to V_p could contain nodes in $R_1, \ldots, R_{m-1}$ for otherwise the path would not be blocked by $A, Q, R_1, \ldots, R_{m-1}$; for the same reason the collider itself could not be in $R_1, \ldots, R_{m-1}$. But it then follows that the collider must itself be one of $V_1, \ldots, V_{p-1}$ since it is an ancestor of V_p with a directed path to V_p not blocked by R. However, if the collider is one of $V_1, \ldots, V_{p-1}$ then the path would in fact be blocked by the parents of the collider since all the parents of $V_1, \ldots, V_{k-1}$ are in $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$. From this it follows that all backdoor paths from R_m to V_k with an edge going out from V_k are blocked by $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$.

We have thus shown that V_k and R_m are d-separated given $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-1}$ and so

$$S(v_k|a, v_1, \ldots, v_{k-1}, q, r) = S(v_k|a, v_1, \ldots, v_{k-1}, q, r_1, \ldots, r_{m-1}).$$

Similarly, V_k and R_{m-1} are d-separated given $A, V_1, \ldots, V_{k-1}, Q, R_1, \ldots, R_{m-2}$ and so

$$S(v_k|a, v_1, \ldots, v_{k-1}, q, r_1, \ldots, r_{m-1}) = S(v_k|a, v_1, \ldots, v_{k-1}, q, r_1, \ldots, r_{m-2}).$$

We may carry this argument forward to get

$$S(v_k|a, v_1, \ldots, v_{k-1}, q, r) = S(v_k|a, v_1, \ldots, v_{k-1}, q, r^k).$$

All backdoor paths from V_k to $Q \backslash Q^k$ will be blocked given $A, V_1, \ldots, V_{k-1}, Q^k, R^k$ by pa_{v_k}. Since V_k is not a descendant of $Q \backslash Q^k$ all frontdoor paths from V_k to $Q \backslash Q^k$ will involve at least one collider which is a descendant of V_k. This collider is not in the conditioning set $A, V_1, \ldots, V_{k-1}, Q^k, R^k$ since this entire set consists of non-descendants of V_k and so the collider will block the frontdoor path from V_k to $Q \backslash Q^k$.

Thus V_k and $Q \backslash Q^k$ are d-separated given $A, V_1, \ldots, V_{k-1}, Q^k, R^k$ and so

$$S(v_k|a, v_1, \ldots, v_{k-1}, q, r^k) = S(v_k|a, v_1, \ldots, v_{k-1}, q^k, r^k).$$

Furthermore, $A, V_1, \ldots, V_{k-1}, Q^k, R^k$ are non-descendants of V_k and include all of the parents of V_k and so

$$S(v_k|a, v_1, \ldots, v_{k-1}, q^k, r^k) = S(v_k|pa_{v_k}).$$

We have thus shown as desired that

$$\begin{aligned} S(v_k|a, v_1, \ldots, v_{k-1}, q, r) &= S(v_k|a, v_1, ..., v_{k-1}, q, r^k) \\ &= S(v_k|a, v_1, ..., v_{k-1}, q^k, r^k) = S(v_k|pa_{v_k}). \end{aligned}$$

Proof of Lemma 11.

Let $V_0 = A$ and $V_n = Y$ and let $V_1, \ldots, V_{n-1}$ be an ordered list of all the nodes which are not in R but which are on directed paths from A to Y such that at least one of the directed paths from each node to Y is not blocked by R. Let $\overline{V}_k = \{V_1, \ldots, V_k\}$. It can be shown by induction that by starting with $n = k$ and for each k iteratively replacing by

their negations the parents of V_k with negative edges into V_k suffices to obtain a graph such that all edges on all directed paths from A to Y not blocked by R have positive sign.

We can express $E[1(V_n > v)|A, Q, R]$ as

$$E[E[\ldots E[E[1(V_n > v)|A, \overline{V}_{n-1}, Q, R]|A, \overline{V}_{n-2}, Q, R]|\ldots|A, V_1, Q, R]|A, Q, R].$$

Now conditional on $A, \overline{V}_{n-1}\backslash V_i, Q, R$ we have that

$$E[1(V_n > v)|, A, \overline{V}_{n-1}, Q, R]$$

is non-decreasing in v_i for $i = 1, \ldots, n-1$ since V_i has either a weak positive monotonic effect or no effect on V_n. Thus conditional on $A, \overline{V}_{n-1}\backslash\{V_i, V_{n-1}\}, Q, R$ we have that

$$E[1(V_n > v)|A, \overline{V}_{n-1}, Q, R]$$

is a non-decreasing function of v_i and v_{n-1}. Furthermore, by Lemma 10 we have that $S(v_{n-1}|a, v_1, \ldots, v_{n-2}, q, r) = S(v_{n-1}|pa_{v_{n-1}})$ and so $S(v_{n-1}|a, v_1, \ldots, v_{n-2}, q, r) = S(v_{n-1}|pa_{v_{n-1}})$ is a non-decreasing in v_i for all $a, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n-2}, q, r$ since V_i has either a weak positive monotonic effect or no effect on V_{n-1}. Thus by Lemma 8 we have that conditional on $A, \overline{V}_{n-2}\backslash V_i, Q, R$,

$$E[E[1(V_n > v)|A, \overline{V}_{n-1}, Q, R]|A, \overline{V}_{n-2}, Q, R]$$

is non-decreasing in v_i for $i = 1, \ldots, n-2$. Carrying the argument forward, conditional on A, Q, R, we will have that

$$E[\ldots E[E[1(V_n > v)|A, \overline{V}_{n-1}, Q, R]|A, \overline{V}_{n-2}, Q, R]|\ldots|A, V_1, Q, R]$$

is a non-decreasing function of v_1 and $v_0 = a$ and since A has either a weak positive monotonic effect or no effect on V_1, $S(v_1|a, q, r) = S(v_1|pa_{v_1})$ will be non-decreasing in a and thus by Lemma 8,

$$\begin{aligned} S(y|a, q, r) &= E[1(V_n > y)|A, Q, R] \\ &= E[E[\ldots E[E[1(V_n > y)|A, \overline{V}_{n-1}, Q, R]|A, \overline{V}_{n-2}, Q, R]|\ldots|A, V_1, Q, R]|A, Q, R] \end{aligned}$$

will be non-decreasing in a.

Proof of Proposition 12

Let Q denote the set of ancestors of A or Y which are not descendants of A. Note that if for each i the backdoor paths from R_i to Y are blocked by $R_1, \ldots, R_{i-1}, A$ and X then these backdoor paths will also be blocked by $R_1, \ldots, R_{i-1}, A$ and Q since for each backdoor path from R_i to X there must be some member of $\{A\} \bigcup Q$ through which the path passes. We may thus apply Lemma 11 to conclude that $E[1(Y > y)|a, Q, r]$. Since Q blocks all backdoor paths from A to Y we have

$$\begin{aligned} S(y|a, x, r) &= E[E[1(Y > y)|a, Q, x, r]|a, x, r] \\ &= E[E[1(Y > y)|a, Q, r]|a, x, r] = E[E[1(Y > y)|a, W, r]|a, x, r] \end{aligned}$$

where W is the subset of Q which are either parents of Y or parents of a node on a directed path from A to Y. Let W' denote the subset of W for which there is a path to Y

not blocked by A, X, R then $E[E[1(Y > y)|a, W, r]|a, x, r] = E[E[1(Y > y)|a, W', r]|a, x, r]$. All backdoor paths from A to W' are blocked given R and X by X since X blocks all backdoor paths from A to Y. Any frontdoor path from A to W' will include a collider since the nodes in W' are not descendants of A. The collider cannot be in X because X includes only non-descendants of A. Suppose the collider were some node R_i; by hypothesis all backdoor paths from R_i to Y are blocked by $R_1, \ldots, R_{i-1}, A$ and X; thus the frontdoor path from A to W' would have to be blocked by $A, R_1, \ldots, R_{i-1}$ and X for otherwise there would be a backdoor path from R_i through W' to Y not blocked by $A, R_1, \ldots, R_{i-1}$ and X. From this it follows that every frontdoor path from A to W' must be blocked given R and X either by a collider or by a node in R or X. We have thus shown that all paths from A to W' are blocked given R and X and so W' is conditionally independent of A given R and X and so we have

$$\begin{aligned} E[E[1(Y > y)|a, W', r]|a, x, r] &= E[E[1(Y > y)|a, W', r]|x, r] \\ &= E[E[1(Y > y)|a, Q, r]|x, r]. \end{aligned}$$

We have thus shown that $S(y|a, x, r) = E[E[1(Y > y)|a, Q, r]|x, r]$. Since $E[1(Y > y)|a, Q, r]$ is non-decreasing in a for all q we also have that

$$S(y|a, x, r) = E[E[1(Y > y)|a, Q, r]|x, r]$$

is non-decreasing in a. Finally, since $S(y|a, x, r)$ is non-decreasing in a, it follows from Lemma 8 that $E[y|a, x, r]$ is also non-decreasing in a.

Proof of Proposition 14.

Proposition 14 is in fact a special case of Proposition 17 with $R = \varnothing$ and $Q = \varnothing$. The proof of Proposition 17 is given below.

Proof of Proposition 15.

Proposition 15 is in fact a special case of Proposition 18 with $R = \varnothing$ and $Q = \varnothing$. The proof of Proposition 18 is given below.

Proof of Proposition 17.

By the law of iterated expectations,

$$\begin{aligned} &E[Y|A = a, Q = q] \\ &\quad = \sum_c E[Y|A = a, C = c, Q = q] P(C = c|A = a, Q = q) \end{aligned}$$

We have by Proposition 12 that $E[Y|A, Q, C]$ is non-decreasing in A. Let $(C_1, \ldots, C_n)$ denote an ordered list of the variables in C. Let Q^c be variables in Q which are common causes of C and let $Q^n = Q \backslash Q^c$. Let Q_i^d be the variables in Q^c that are descendants of C_i. Let C_i^d denote the variables in C that are descendants of C_i and let $C_i^n = C \backslash \{C_i, C_i^d\}$. By Proposition 12 we have that $E[Y|A, Q, C]$ is non-decreasing in each component C_i of C by choosing for each i, A in Proposition 12 to be C_i, X in Proposition 12 to be the set $\{Q^n, Q^c \backslash Q_i^d, C_i^n\}$ and R in Proposition 12 to be the set $\{Q_i^d, C_i^d, A\}$. Furthermore,

$$P(C = c|A = a, Q = q) = \frac{P(A = a|C = c, Q = q) P(C = c|Q = q)}{P(A = a|Q = q)}$$

and so

$$P(C = c|A = 1, Q = q) = \nu_q(c)P(C = c|A = 0, Q = q)$$

where

$$\nu_q(c) = \frac{P(A = 0|Q = q)P(A = 1|C = c, Q = q)}{P(A = 1|Q = q)P(A = 0|C = c, Q = q)}$$

which is non-decreasing in each dimension of c since the numerator is non-decreasing in each dimension of c and the denominator is non-increasing in each dimension of c by Proposition 12 by choosing for each i, A in Proposition 12 to be C_i, X in Proposition 12 to be the set $\{Q^n, Q^c \backslash Q_i^c, C_i^n\}$ and R in Proposition 12 to be the set $\{Q_i^c, C_i^d\}$. Thus

$$\begin{aligned}
&E[Y|A = 1, Q = q] \\
&= \sum_c E[Y|A = 1, C = c, Q = q]P(C = c|A = 1, Q = q) \\
&\geq \sum_c E[Y|A = 0, C = c, Q = q]P(C = c|A = 1, Q = q) \\
&= \sum_c E[Y|A = 0, C = c, Q = q]\nu_q(c)P(C = c|A = 0, Q = q) \\
&\geq \sum_c E[Y|A = 0, C = c, Q = q]P(C = c|A = 0, Q = q) \\
&= E[Y|A = 0, Q = q].
\end{aligned}$$

The second inequality holds because by an argument similar to that above $E[Y|A = 0, Q = q, C = c]$ is non-decreasing in each dimension of c and $P(C = c|A = 1, Q = q) = \nu_q(c)P(C = c|A = 0, Q = q)$ weights more heavily higher values of each dimension of c than does $P(C = c|A = 0, Q = q)$ since $\nu_q(c)$ is non-decreasing in each dimension of c. Thus $E[Y|A = a, Q = q]$ is non-decreasing in a.

Proof of Proposition 18.

By the law of iterated expectations we have that

$$\begin{aligned}
&E[A|Y = y, Q = q] = \sum_c E[A|Y = y, C = c, Q = q]P(C = c|Y = y, Q = q) \\
&= \sum_{c,a} aP(A = a|Y = y, C = c, Q = q)P(C = c|Y = y, Q = q) \\
&= \sum_{c,a} a\frac{P(Y = y, A = a, C = c|Q = q)}{P(Y = y, C = c|Q = q)}P(C = c|Y = y, Q = q) \\
&= \sum_{c,a} a\frac{P(Y = y|A = a, C = c, Q = q)}{P(Y = y|Q = q)}P(A = a, C = c|Q = q) \\
&= E_{C,A}[A\frac{P(Y = y|A, C, Q = q)}{P(Y = y|Q = q)}|Q = q].
\end{aligned}$$

As in the proof of Proposition 17, we have by Proposition 12 we have that conditional on and $Q = q$, $\frac{P(Y=1|A,C,Q=q)}{P(Y=1|Q=q)}$ is a non-decreasing function of A and of each dimension of C. Similarly, $\frac{P(Y=0|A,C,Q=q)}{P(Y=0|Q=q)}$ is a non-increasing function of A and each dimension of C. Over c and a, conditional on and $Q = q$, $\frac{P(Y=y|A=a,C=c,Q=q)}{P(Y=y|Q=q)}$ is a weight function that sums to 1 i.e. $E_{C,A}[\frac{P(Y=y|A=a,C=c,Q=q)}{P(Y=y|Q=q)}] = \frac{P(Y=y|Q=q)}{P(Y=y|Q=q)} = 1$. Furthermore, by

Proposition 12, $S(a|c,q)$ is non-decreasing in c and we thus have that

$$\begin{aligned} E[A|Y=1,Q=q] &= E_{C,A}[A\frac{P(Y=1|A,C,Q=q)}{P(Y=1|Q=q)}|Q=q] \\ &\geq E_{C,A}[A\frac{P(Y=0|A,C,Q=q)}{P(Y=0|Q=q)}|Q=q] \\ &= E[A|Y=0,Q=q] \end{aligned}$$

and so $E[A|Y,Q]$ is non-decreasing in Y.

Proof of Proposition 21.

Note that by Proposition 9 above if A_1 has a weak positive monotonic effect on Y then $E[Y|A_1 = a_1, A_2 = a_2, C = c]$ must be non-decreasing in a_1 and if A_1 has a weak negative monotonic effect on Y then $E[Y|A_1 = a_1, A_2 = a_2, C = c]$ must be non-increasing in a_1. Since $(Y \coprod A_1|\{A_2, C\})_{G_{\underline{E_1}}}$ where $G_{\underline{E_1}}$ is the original directed acyclic graph G with all edges emanating from A_1 removed, we have $Y_{A_1=a} \coprod A_1|\{A_2, C\}$ (Pearl, 1995). Thus $E[Y_{A_1=a_1}|A_2 = a_2, C = c] = E[Y|A_1 = a_1, A_2 = a_2, C = c]$ and so if A_2 is a qualitative effect modifier for the causal effect of A_1 on Y for stratum $C = c$ then we must two values of A_1, a_1^* and a_1^{**}, and two levels of A_2, a_2' and a_2'', such that $E[Y|A_1 = a_1^{**}, A_2 = a_2'', C = c] - E[Y|A_1 = a_1^*, A_2 = a_2'', C = c] < 0$ and $E[Y|A_1 = a_1^{**}, A_2 = a_2', C = c] - E[Y|A_1 = a_1^*, A_2 = a_2', C = c] > 0$. Either $a_1^{**} > a_1^*$ or $a_1^{**} < a_1^*$. Consider the first case (the second is analogous) then since $E[Y|A_1 = a_1^{**}, A_2 = a_2'', C = c] - E[Y|A_1 = a_1^*, A_2 = a_2'', C = c] < 0$, A_1 does not have a weak positive monotonic effect on Y and since $E[Y|A_1 = a_1^{**}, A_2 = a_2', C = c] - E[Y|A_1 = a_1^*, A_2 = a_2', C = c] > 0$, A_1 does not have a weak negative monotonic effect on Y. Now if A_2 is a qualitative effect modifier for the causal effect of A_1 unconditionally then we must have two values of A_1, a_1^* and a_1^{**}, and two levels of A_2, a_2' and a_2'', such that $E[Y_{A_1=a_1^{**}}|A_2 = a_2''] - E[Y_{A_1=a_1^*}|A_2 = a_2''] < 0$ and $E[Y_{A_1=a_1^{**}}|A_2 = a_2'] - E[Y_{A_1=a_1^*}|A_2 = a_2'] > 0$. Once again either $a_1^{**} > a_1^*$ or $a_1^{**} < a_1^*$. We will consider the first case (the second is analogous). We thus have that $\sum_c E[Y|A_1 = a_1^{**}, A_2 = a_2'', C = c]P(C = c|A_2 = a_2'') = \sum_c E[Y_{A_1=a_1^{**}}|A_2 = a_2'', C = c]P(C = c|A_2 = a_2'') = E[Y_{A_1=a_1^{**}}|A_2 = a_2''] < E[Y_{A_1=a_1^*}|A_2 = a_2''] = \sum_c E[Y_{A_1=a_1^*}|A_2 = a_2'', C = c]P(C = c|A_2 = a_2'') = \sum_c E[Y|A_1 = a_1^*, A_2 = a_2'', C = c]P(C = c|A_2 = a_2'')$ and so A_1 cannot have a weak positive monotonic effect on Y and similarly, $\sum_c E[Y|A_1 = a_1^{**}, A_2 = a_2', C = c]P(C = c|A_2 = a_2') = \sum_c E[Y_{A_1=a_1^{**}}|A_2 = a_2', C = c]P(C = c|A_2 = a_2') = E[Y_{A_1=a_1^{**}}|A_2 = a_2'] > E[Y_{A_1=a_1^*}|A_2 = a_2'] = \sum_c E[Y_{A_1=a_1^*}|A_2 = a_2', C = c]P(C = c|A_2 = a_2') = \sum_c E[Y|A_1 = a_1^*, A_2 = a_2', C = c]P(C = c|A_2 = a_2')$ and so A_1 cannot have a weak negative monotonic effect on Y.

Proof of Proposition 23.

We prove the Theorem for weak positive monotonic effects. The proof for weak negative monotonic effects is similar. Let C denote all non-descendants of A which are either parents of Y or parents of a node on a directed path between A and Y. By the law of iterated expectations we have $E[Y_{A=a_1}|Q = q] - E[Y_{A=a_0}|Q = q] = \sum_c E[Y_{A=a_1}|C =$

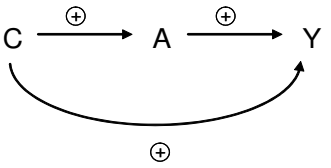


Figure 5: Directed acyclic graph illustrating counterexamples to Propositions 14 and 15 when A is not binary.

$c, Q = q]P(C = c|Q = q) - \sum_c E[Y_{A=a_0}|C = c, Q = q]P(C = c|Q = q)$. We will show that this latter expression is equal to $\sum_c E[Y_{A=a_1}|C = c]P(C = c|Q = q) - \sum_c E[Y_{A=a_0}|C = c]P(C = c|Q = q)$. By Theorem 3 of Pearl (1995) it suffices to show that $(Y \coprod Q|C, A)_{G_{\overline{A}}}$ where $G_{\overline{A}}$ denotes the graph obtained by deleting from the original directed acyclic graph all arrows pointing into A. Any front door path from Y to Q in $G_{\overline{A}}$ will be blocked by a collider. Any backdoor path from Y to Q in $G_{\overline{A}}$ will be blocked by C. We thus have that $E[Y_{A=a_1}|Q = q] - E[Y_{A=a_0}|Q = q] = \sum_c E[Y_{A=a_1}|C = c]P(C = c|Q = q) - \sum_c E[Y_{A=a_0}|C = c]P(C = c|Q = q)$. Since C will block all backdoor paths from A to Y we have by the backdoor path adjustment theorem $\sum_c E[Y|C = c, A = a_1]P(C = c|Q = q) - \sum_c E[Y|C = c, A = a_0]P(C = c|Q = q) = \sum_c \{E[Y|C = c, A = a_1] - E[Y|C = c, A = a_0]\}P(C = c|Q = q)$. If there were a qualitative effect modifier Q for the causal effect of A on Y then there would exist a value q_0 such that $E[Y_{A=a_1}|Q = q_0] - E[Y_{A=a_0}|Q = q_0] < 0$. But since all paths between A and Y are of positive sign and since C blocks all backdoor paths from A to Y we have by Proposition 12 that $E[Y|C = c, A = a]$ is non-decreasing in a and so $E[Y_{A=a_1}|Q = q_0] - E[Y_{A=a_0}|Q = q_0] = \sum_c \{E[Y|C = c, A = a_1] - E[Y|C = c, A = a_0]\}P(C = c|Q = q_0) \geq 0$.

Appendix B. Counterexamples.

Counterexample 1
Consider the directed acyclic graph given in Figure 5.

In this example C and Y are binary and A is ternary. Suppose that $C \sim Ber(0.5)$, $\epsilon_A \sim Ber(0.5)$ and that $P(A = 0|\epsilon_A = 0) = 1$ and $P(A = C + 1|\epsilon_A = 1) = 1$. Suppose also that $P(Y = 1|A = 2) = 1$ and that if $P(Y = C|A = 0) = 1$ and $P(Y = C|A = 1) = 1$. Clearly then C has a positive monotonic effect on A and on Y and A has a positive monotonic effect on Y and so A and Y are positively monotonically associated. However, we have that $E[Y|A = 1] = E[C|A = 1] = 0 * P(C = 1|A = 1) = 0$ but $E[Y|A = 0] = E[C|A = 0] = 1 * P(C = 1|A = 0) + 0 * P(C = 0|A = 0) = 1/2$.

Counterexample 2
Consider again the directed acyclic graph given in Figure 5. In this example we will assume that C and A are binary and that Y is ternary. Suppose that $C \sim Ber(0.5)$ and that ϵ_A takes on the values 0, 1 and 2, each with probability 1/3. Suppose also that $P(A = 0|\epsilon_A = 0) = 1$, $P(A = C|\epsilon_A = 1) = 1$ and $P(A = 1|\epsilon_A = 2) = 1$. Suppose further that $P(Y = 0|C = 0) = 1$ and if $P(Y = A + 1|C = 1)$. Clearly then C has a positive monotonic effect on A and on Y and A has a positive monotonic effect on Y and so A and Y are positively monotonically associated. However, we have that $E[A|Y = 1] = 0$ but $E[A|Y = 0] = E[A|C = 0] = 1/3$.

References

Alexander Philip Dawid. Influence diagrams for causal modelling and inference. *Int. Statist. Rev.*, 70:161–189, 2002.

Marek J. Druzdzel and Max Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the National Conference on Artificial Intelligence*, pages 548–553, Washington D. C., 1993.

Dan Geiger, Thomas S. Verma, and Judea Pearl. Identifying independence in bayesian networks. *Networks*, 20:507–534, 1990.

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

Wolfgang Härdle, Gerard Kerkyacharian, Dominique Picard, and Alexander Tsybakov. *Wavelets, Approximation, and Statistical Applications*. Springer Verlag, New York, 1998.

Changsung Kang and Jin Tian. Inequality constraints in causal models with hidden variables. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 233–240, Cambridge, 2006. AUAI Press.

Changsung Kang and Jin Tian. Polynomial constraints in causal bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 200–208, Cambridge, 2007. AUAI Press.

Sefan L. Lauritzen, Alexander Philip Dawid, B. N. Larsen, and H. G. Leimer. Independence properties of directed markov fields. *Networks*, 20:491–505, 1990.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82:669–688, 1995.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Conference on Uncertainty and Artificial Intelligence*, pages 411–420, San Francisco, 2001. Morgan Kaufmann.

James M. Robins. Semantics of causal dag models and the identification of direct and indirect effects. In P. Green, N.L. Hjort, and S. Richardson, editors, *Highly Structured Stochastic Systems*, pages 70–81. Oxford University Press, New York, 2003.

James M. Robins and Sander Greenland. Identifiability and exchangeability for direct and indirect effects. *Epidemiol.*, 3:143–155, 1992.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction and Search*. Springer-Verlag, New York, 2000.

Jin Tian and Judea Pearl. On the testable implications of causal models with hidden variables. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 519–527, San Francisco, 2002. Morgan Kaufmann.

Linda C. van der Gaag, Hans L. Bodlaender, and Ad Feelders. Monotonicity in bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 519–527, Banff Canada, 2004. AUAI Press.

Tyler J. VanderWeele and James M. Robins. Four types of effect modification, a classification based on directed acyclic graphs. *Epidemiol.*, 18:561–568, 2007.

Tyler J. VanderWeele and James M. Robins. Empirical and counterfactual conditions for sufficient cause interactions. *Biometrika*, 95:49–61, 2008.

Tyler J. VanderWeele and James M. Robins. Signed directed acyclic graphs for causal inference. *Journal of the Royal Statistical Society Series B*, in press, 2009.

Thomas S. Verma and Judea Pearl. Causal networks: Semantics and expressiveness. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 4, pages 352–359, 1988.

Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

Sewall Wright. Correlation and causation. *J. Agric. Res.*, 20:557–585, 1921.

JMLR: Workshop and Conference Proceedings 10:1527–1570, 2009 Submitted 7/07; Revised 3/08; Published 7/09

Bayesian Network Structure Learning by Recursive Autonomy Identification

Raanan Yehezkel RAANAN.YEHEZKEL@GMAIL.COM
Video Analytics Group
NICE Systems Ltd.
8 Hapnina, POB 690, Raanana, 43107, Israel

Boaz Lerner BOAZ@BGU.AC.IL
Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer-Sheva, 84105, Israel

Editor: Constantin Aliferis

Abstract

We propose the recursive autonomy identification (RAI) algorithm for constraint-based (CB) Bayes-ian network structure learning. The RAI algorithm learns the structure by sequential application of conditional independence (CI) tests, edge direction and structure decomposition into autonomous sub-structures. The sequence of operations is performed recursively for each autonomous sub-structure while simultaneously increasing the order of the CI test. While other CB algorithms d-separate structures and then direct the resulted undirected graph, the RAI algorithm combines the two processes from the outset and along the procedure. By this means and due to structure decomposition, learning a structure using RAI requires a smaller number of CI tests of high orders. This reduces the complexity and run-time of the algorithm and increases the accuracy by diminishing the curse-of-dimensionality. When the RAI algorithm learned structures from databases representing synthetic problems, known networks and natural problems, it demonstrated superiority with respect to computational complexity, run-time, structural correctness and classification accuracy over the PC, Three Phase Dependency Analysis, Optimal Reinsertion, greedy search, Greedy Equivalence Search, Sparse Candidate, and Max-Min Hill-Climbing algorithms.

Keywords: Bayesian networks, constraint-based structure learning

1. Introduction

A Bayesian network (BN) is a graphical model that efficiently encodes the joint probability distribution for a set of variables (Heckerman, 1995; Pearl, 1988). The BN consists of a structure and a set of parameters. The structure is a directed acyclic graph (DAG) that is composed of nodes representing domain variables and edges connecting these nodes. An edge manifests dependence between the nodes connected by the edge, while the absence of an edge demonstrates independence between the nodes. The parameters of a BN are conditional probabilities (densities) that quantify the graph edges. Once the BN structure has been learned, the parameters are usually estimated (in the case of discrete variables) using the relative frequencies of all combinations of variable states as exemplified in the data. Learning the structure from data by considering all possible structures exhaustively is not feasible in most domains, regardless of the size of the data

(Chickering et al., 2004), since the number of possible structures grows exponentially with the number of nodes (Cooper and Herskovits, 1992). Hence, structure learning requires either sub-optimal heuristic search algorithms or algorithms that are optimal under certain assumptions.

One approach to structure learning — known as search-and-score (S&S) (Chickering, 2002; Cooper and Herskovits, 1992; Heckerman, 1995; Heckerman et al., 1995) — combines a strategy for searching through the space of possible structures with a scoring function measuring the fitness of each structure to the data. The structure achieving the highest score is then selected. Algorithms of this approach may also require node ordering, in which a parent node precedes a child node so as to narrow the search space (Cooper and Herskovits, 1992). In a second approach — known as constraint-based (CB) (Cheng et al., 1997; Pearl, 2000; Spirtes et al., 2000) — each structure edge is learned if meeting a constraint usually derived from comparing the value of a statistical or information-theory-based test of conditional independence (CI) to a threshold. Meeting such constraints enables the formation of an undirected graph, which is then further directed based on orientation rules (Pearl, 2000; Spirtes et al., 2000). That is, generally in the S&S approach we learn structures, whereas in the CB approach we learn edges composing a structure.

Search-and-score algorithms allow the incorporation of user knowledge through the use of prior probabilities over the structures and parameters (Heckerman et al., 1995). By considering several models altogether, the S&S approach may enhance inference and account better for model uncertainty (Heckerman et al., 1999). However, S&S algorithms are heuristic and usually have no proof of correctness (Cheng et al., 1997) (for a counter-example see Chickering, 2002, providing an S&S algorithm that identifies the optimal graph in the limit of a large sample and has a proof of correctness). As mentioned above, S&S algorithms may sometimes depend on node ordering (Cooper and Herskovits, 1992). Recently, it was shown that when applied to classification, a structure having a higher score does not necessarily provide a higher classification accuracy (Friedman et al., 1997; Grossman and Domingos, 2004; Kontkanen et al., 1999).

Algorithms of the CB approach are generally asymptotically correct (Cheng et al., 1997; Spirtes et al., 2000). They are relatively quick and have a well-defined stopping criterion (Dash and Druzdzel, 2003). However, they depend on the threshold selected for CI testing (Dash and Druzdzel, 1999) and may be unreliable in performing CI tests using large condition sets and a limited data size (Cooper and Herskovits, 1992; Heckerman et al., 1999; Spirtes et al., 2000). They can also be unstable in the sense that a CI test error may lead to a sequence of errors resulting in an erroneous graph (Dash and Druzdzel, 1999; Heckerman et al., 1999; Spirtes et al., 2000). Additional information on the above two approaches, their advantages and disadvantages, may be found in Cheng et al. (1997), Cooper and Herskovits (1992), Dash and Druzdzel (1999), Dash and Druzdzel (2003), Heckerman (1995), Heckerman et al. (1995), Heckerman et al. (1999), Pearl (2000) and Spirtes et al. (2000). We note that Cowell (2001) showed that for complete data, a given node ordering and using cross-entropy methods for checking CI and maximizing logarithmic scores to evaluate structures, the two approaches are equivalent. In addition, hybrid algorithms have been suggested in which a CB algorithm is employed to create an initial ordering (Singh and Valtorta, 1995), to obtain a starting graph (Spirtes and Meek, 1995; Tsamardinos et al., 2006a) or to narrow the search space (Dash and Druzdzel, 1999) for an S&S algorithm.

Most CB algorithms, such as Inductive Causation (IC) (Pearl, 2000), PC (Spirtes et al., 2000) and Three Phase Dependency Analysis (TPDA) (Cheng et al., 1997), construct a

DAG in two consecutive stages. The first stage is learning associations between variables for constructing an undirected structure. This requires a number of CI tests growing exponentially with the number of nodes. This complexity is reduced in the PC algorithm to polynomial complexity by fixing the maximal number of parents a node can have and in the TPDA algorithm by measuring the strengths of the independences computed while CI testing along with making a strong assumption about the underlying graph (Cheng et al., 1997). The TPDA algorithm does not take direct steps to restrict the size of the condition set employed in CI testing in order to mitigate the curse-of-dimensionality.

In the second stage, most CB algorithms direct edges by employing orientation rules in two consecutive steps: finding and directing V-structures and directing additional edges inductively (Pearl, 2000). Edge direction (orientation) is unstable. This means that small errors in the input to the stage (i.e., CI testing) yield large errors in the output (Spirtes et al., 2000). Errors in CI testing are usually the result of large condition sets. These sets, selected based on previous CI test results, are more likely to be incorrect due to their size, and they also lead, for a small sample size, to poorer estimation of dependences due to the curse-of-dimensionality. Thus, we usually start learning using CI tests of low order (i.e., using small condition sets), which are the most reliable tests (Spirtes et al., 2000). We further note that the division of learning in CB algorithms into two consecutive stages is mainly for simplicity, since no directionality constraints have to be propagated during the first stage. However, errors in CI testing is a main reason for the instability of CB algorithms, which we set out to tackle in this research.

We propose the recursive autonomy identification (RAI) algorithm, which is a CB model that learns the structure of a BN by sequential application of CI tests, edge direction and structure decomposition into autonomous sub-structures that comply with the Markov property (i.e., the sub-structure includes all its nodes' parents). This sequence of operations is performed recursively for each autonomous sub-structure. In each recursive call of the algorithm, the order of the CI test is increased similarly to the PC algorithm (Spirtes et al., 2000). By performing CI tests of low order (i.e., tests employing small conditions sets) before those of high order, the RAI algorithm performs more reliable tests first, and thereby obviates the need to perform less reliable tests later. By directing edges while testing conditional independence, the RAI algorithm can consider parent-child relations so as to rule out nodes from condition sets and thereby to avoid unnecessary CI tests and to perform tests using smaller condition sets. CI tests using small condition sets are faster to implement and more accurate than those using large sets. By decomposing the graph into autonomous sub-structures, further elimination of both the number of CI tests and size of condition sets is obtained. Graph decomposition also aids in subsequent iterations to direct additional edges. By recursively repeating both mechanisms for autonomies decomposed from the graph, further reduction of computational complexity, database queries and structural errors in subsequent iterations is achieved. Overall, the RAI algorithm learns faster a more precise structure.

Tested using synthetic databases, nineteen known networks, and nineteen UCI databases, RAI showed in this study superiority with respect to structural correctness, complexity, run-time and classification accuracy over PC, Three Phase Dependency Analysis, Optimal Reinsertion, a greedy hill-climbing search algorithm with a Tabu list, Greedy Equivalence Search, Sparse Candidate, naive Bayesian, and Max-Min Hill-Climbing algorithms.

After providing some preliminaries and definitions in Section 2, we introduce the RAI algorithm and prove its correctness in Section 3. Section 4 presents experimental

evaluation of the RAI algorithm with respect to structural correctness, complexity, runtime and classification accuracy in comparison to CB, S&S and hybrid structure learning algorithms. Section 5 concludes the paper with a discussion.

2. Preliminaries

A BN $B(\mathcal{G}, \Theta)$ is a model for representing the joint probability distribution for a set of variables $\boldsymbol{X} = \{X_1 \dots X_n\}$. The structure $\mathcal{G}(\boldsymbol{V}, \boldsymbol{E})$ is a DAG composed of $\boldsymbol{V}$, a set of nodes representing the domain variables $\boldsymbol{X}$, and $\boldsymbol{E}$, a set of directed edges connecting the nodes. A directed edge $X_i \rightarrow X_j$ connects a child node X_j to its parent node X_i. We denote $\boldsymbol{Pa}(X, \mathcal{G})$ as the set of parents of node X in a graph $\mathcal{G}$. The set of parameters Θ holds local conditional probabilities over $\boldsymbol{X}$, $P(X_i|\boldsymbol{Pa}(X_i, \mathcal{G}))\forall i$ that quantify the graph edges. The joint probability distribution for $\boldsymbol{X}$ represented by a BN that is assumed to encode this distribution[1] is (Cooper and Herskovits, 1992; Heckerman, 1995; Pearl, 1988)

$$P(X_1 \dots X_n) = \prod_{i=1}^{n} P(X_i|\boldsymbol{Pa}(X_i, \mathcal{G})). \tag{1}$$

Though there is no theoretical restriction on the functional form of the conditional probability distributions in Equation 1, we restrict ourselves in this study to discrete variables. This implies joint distributions which are unrestricted discrete distributions and conditional probability distributions which are independent multinomials for each variable and each parent configuration (Chickering, 2002).

We also make use of the term partially directed graph, that is, a graph that may have both directed and undirected edges and has at most one edge between any pair of nodes (Meek, 1995). We use this term while learning a graph starting from a complete undirected graph and removing and directing edges until uncovering a graph representing a family of Markov equivalent structures (pattern) of the true underlying BN[2] (Pearl, 2000; Spirtes et al., 2000). $\boldsymbol{Pa}_p(X, \mathcal{G})$, $\boldsymbol{Adj}(X, \mathcal{G})$ and $\boldsymbol{Ch}(X, \mathcal{G})$ are, respectively, the sets of potential parents, adjacent nodes[3] and children of node X in a partially directed graph $\mathcal{G}$, $\boldsymbol{Pa}_p(X, \mathcal{G}) = \boldsymbol{Adj}(X, \mathcal{G}) \backslash \boldsymbol{Ch}(X, \mathcal{G})$.

We indicate that X and Y are independent conditioned on a set of nodes $\boldsymbol{S}$ (i.e., the condition set) using $X \perp\!\!\!\perp Y \,|\, \boldsymbol{S}$, and make use of the notion of d-separation (Pearl, 1988). Thereafter, we define d-separation resolution with the aim to evaluate d-separation for different sizes of condition sets, d-separation resolution of a graph, an exogenous cause to a graph and an autonomous sub-structure. We concentrate in this section only on terms and definitions that are directly relevant to the RAI concept and algorithm, where other more general terms and definitions relevant to BNs can be found in Heckerman (1995), Pearl (1988), Pearl (2000), and Spirtes et al. (2000).

Definition 1 (d-separation resolution) *The resolution of a d-separation relation between a pair of non-adjacent nodes in a graph is the size of the smallest condition set that d-separates the two nodes.*

Examples of d-separation resolutions of 0, 1 and 2 between nodes X and Y are given in Figure 1.

1. Throughout the paper, we assume faithfulness of the probability distribution to a DAG (Spirtes et al., 2000).
2. Two BNs are Markov equivalent if and only if they have the same sets of adjacencies and V-structures (Verma and Pearl, 1990).
3. Two nodes in a graph that are connected by an edge are adjacent.

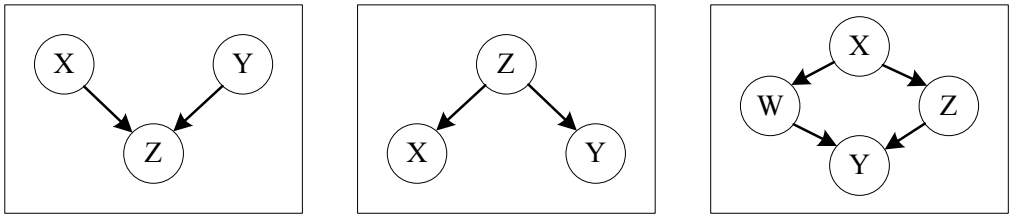


Figure 1: Examples of d-separation resolutions of (a) 0, (b) 1 and (c) 2 between nodes X and Y.

Definition 2 (d-separation resolution of a graph) *The d-separation resolution of a graph is the highest d-separation resolution in the graph.*

The d-separation relations encoded by the example graph in Figure 2a and relevant to the determination of the d-separation resolution of this graph are: 1) $X_1 \perp\!\!\!\perp X_2 \,|\, \emptyset$; 2) $X_1 \perp\!\!\!\perp X_4 \,|\, \{X_3\}$; 3) $X_1 \perp\!\!\!\perp X_5 \,|\, \{X_3\}$; 4) $X_1 \perp\!\!\!\perp X_6 \,|\, \{X_3\}$; 5) $X_2 \perp\!\!\!\perp X_4 \,|\, \{X_3\}$; 6) $X_2 \perp\!\!\!\perp X_5 \,|\, \{X_3\}$; 7) $X_2 \perp\!\!\!\perp X_6 \,|\, \{X_3\}$; 8) $X_3 \perp\!\!\!\perp X_6 \,|\, \{X_4, X_5\}$ and 9) $X_4 \perp\!\!\!\perp X_5 \,|\, \{X_3\}$. Due to relation 8, exemplifying d-separation resolution of 2, the d-separation resolution of the graph is 2. Eliminating relation 8 by adding the edge $X_3 \rightarrow X_6$, we form a graph having a d-separation resolution of 1 (Figure 2b). By further adding edges to the graph, eliminating relations of resolution 1, we form a graph having a d-separation resolution of 0 (Figure 2c) that encodes only relation 1.

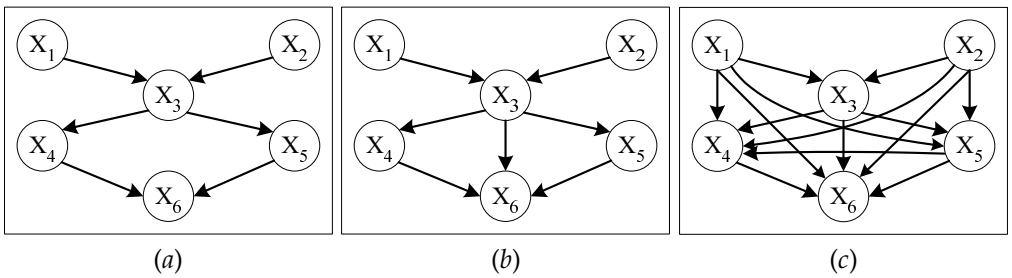


Figure 2: Examples of graph d-separation resolutions of (a) 2, (b) 1 and (c) 0.

Definition 3 (exogenous cause) *A node Y in $\mathcal{G}(\mathbf{V}, \mathbf{E})$ is an exogenous cause to $\mathcal{G}'(\mathbf{V}', \mathbf{E}')$, where $\mathbf{V}' \subset \mathbf{V}$ and $\mathbf{E}' \subset \mathbf{E}$, if $Y \notin \mathbf{V}'$ and $\forall X \in \mathbf{V}'$, $Y \in \mathbf{Pa}(X, \mathcal{G})$ or $Y \notin \mathbf{Adj}(X, \mathcal{G})$ (Pearl, 2000).*

Definition 4 (autonomous sub-structure) *In a DAG $\mathcal{G}(\mathbf{V}, \mathbf{E})$, a sub-structure $\mathcal{G}^{\mathrm{A}}(\mathbf{V}^{\mathrm{A}}, \mathbf{E}^{\mathrm{A}})$ such that $\mathbf{V}^{\mathrm{A}} \subset \mathbf{V}$ and $\mathbf{E}^{\mathrm{A}} \subset \mathbf{E}$ is said to be autonomous in $\mathcal{G}$ given a set $\mathbf{V}_{ex} \subset \mathbf{V}$ of exogenous causes to $\mathcal{G}^{\mathrm{A}}$ if $\forall X \in \mathbf{V}^{\mathrm{A}}$, $\mathbf{Pa}(X, \mathcal{G}) \subset \{\mathbf{V}^{\mathrm{A}} \cup \mathbf{V}_{ex}\}$. If $\mathbf{V}_{ex}$ is empty, we say the sub-structure is (completely) autonomous*[4].

4. If $\mathcal{G}$ is a partially directed graph, then $\boldsymbol{Pa}_p(X, \mathcal{G})$ replaces $\boldsymbol{Pa}(X, \mathcal{G})$.

We define sub-structure autonomy in the sense that the sub-structure holds the Markov property for its nodes. Given a structure $\mathcal{G}$, any two non-adjacent nodes in an autonomous sub-structure $\mathcal{G}^{\mathrm{A}}$ in $\mathcal{G}$ are d-separated given nodes either included in the sub-structure $\mathcal{G}^{\mathrm{A}}$ or exogenous causes to $\mathcal{G}^{\mathrm{A}}$. Figure 3 depicts a structure $\mathcal{G}$ containing a sub-structure $\mathcal{G}^{\mathrm{A}}$. Since nodes X_1 and X_2 are exogenous causes to $\mathcal{G}^{\mathrm{A}}$ (i.e., they are either parents of nodes in $\mathcal{G}^{\mathrm{A}}$ or not adjacent to them; see Definition 3), $\mathcal{G}^{\mathrm{A}}$ is said to be autonomous in $\mathcal{G}$ given nodes X_1 and X_2.

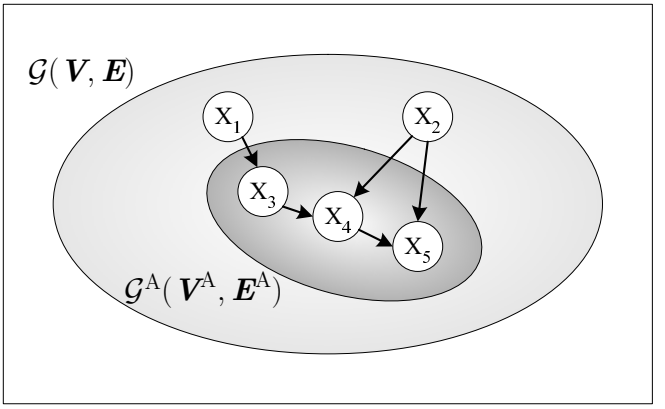


Figure 3: An example of an autonomous sub-structure.

Proposition 5 *If $\mathcal{G}^{\mathrm{A}}(\mathbf{V}^{\mathrm{A}}, \mathbf{E}^{\mathrm{A}})$ is an autonomous sub-structure in a DAG $\mathcal{G}(\mathbf{V}, \mathbf{E})$ given a set $\mathbf{V}_{ex} \subset \mathbf{V}$ of exogenous causes to $\mathcal{G}^{\mathrm{A}}$ and $X \perp\!\!\!\perp Y \mid \mathbf{S}$, where $X, Y \in \mathbf{V}^{\mathrm{A}}$, $\mathbf{S} \subset \mathbf{V}$, then $\exists \mathbf{S}'$ such that $\mathbf{S}' \subset \{\mathbf{V}^{\mathrm{A}} \cup \mathbf{V}_{ex}\}$ and $X \perp\!\!\!\perp Y \mid \mathbf{S}'$.*

Proof The proof is based on Lemma 6. ■

Lemma 6 *If in a DAG, X and Y are non-adjacent and X is not a descendant of Y,[5] then X and Y are d-separated given $\mathbf{Pa}(Y)$ (Pearl, 1988; Spirtes et al., 2000).*

If in a DAG $\mathcal{G}(\boldsymbol{V}, \boldsymbol{E})$, $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$ for some set $\boldsymbol{S}$, where X and Y are non-adjacent, and if X is not a descendant of Y, then, according to Lemma 6, X and Y are d-separated given $\boldsymbol{Pa}(Y)$. Since X and Y are contained in the sub-structure $\mathcal{G}^{\mathrm{A}}(\boldsymbol{V}^{\mathrm{A}}, \boldsymbol{E}^{\mathrm{A}})$, which is autonomous given the set of nodes $\boldsymbol{V}_{ex}$, then, following the definition of an autonomous sub-structure, all parents of the nodes in $\boldsymbol{V}^{\mathrm{A}}$ — and specifically $\boldsymbol{Pa}(Y)$ — are members in set $\{\boldsymbol{V}^{\mathrm{A}} \cup \boldsymbol{V}_{ex}\}$. Then, $\exists \boldsymbol{S}'$ such that $\boldsymbol{S}' \subset \{\boldsymbol{V}^{\mathrm{A}} \cup \boldsymbol{V}_{ex}\}$ and $X \perp\!\!\!\perp Y \mid \boldsymbol{S}'$, which proves Proposition 5. ■

3. Recursive Autonomy Identification

Starting from a complete undirected graph and proceeding from low to high graph d-separation resolution, the RAI algorithm uncovers the correct pattern[6] of a structure

5. If X is a descendant of Y, we change the roles of X and Y and replace $\boldsymbol{Pa}(Y)$ with $\boldsymbol{Pa}(X)$.
6. In the absence of a topological node ordering, uncovering the correct pattern is the ultimate goal of BN structure learning algorithms, since a pattern represents the same set of probabilities as that of the true structure (Spirtes et al., 2000).

by performing the following sequence of operations: (1) test of CI between nodes, followed by the removal of edges related to independences, (2) edge direction according to orientation rules, and (3) graph decomposition into autonomous sub-structures. For each autonomous sub-structure, the RAI algorithm is applied recursively, while increasing the order of CI testing.

CI testing of order n between nodes X and Y is performed by thresholding the value of a criterion that measures the dependence between the nodes conditioned on a set of n nodes (i.e., the condition set) from the parents of X or Y. The set is determined by the Markov property (Pearl, 2000), for example, if X is directed into Y, then only Y's parents are included in the set. Commonly, this criterion is the χ^2 goodness of fit test (Spirtes et al., 2000) or conditional mutual information (CMI) (Cheng et al., 1997).

Directing edges is conducted according to orientation rules (Pearl, 2000; Spirtes et al., 2000). Given an undirected graph and a set of independences, both being the result of CI testing, the following two steps are performed consecutively. First, intransitive triplets of nodes (V-structures) are identified, and the corresponding edges are directed. An intransitive triplet $X \rightarrow Z \leftarrow Y$ is defined if 1) X and Y are non-adjacent neighbors of Z, and 2) Z is not in the condition set that separated X and Y. In the second step, also known as the inductive stage, edges are continually directed until no more edges can be directed, while assuring that no new V-structures and no directed cycles are created.

Decomposition into separated, smaller, autonomous sub-structures reveals the structure hierarchy. Decomposition also decreases the number and length of paths between nodes that are CI-tested, thereby diminishing, respectively, the number of CI tests and the sizes of condition sets used in these tests. Both reduce computational complexity. Moreover, due to decomposition, additional edges can be directed, which reduces the complexity of CI testing of the subsequent iterations. Following decomposition, the RAI algorithm identifies ancestor and descendant sub-structures; the former are autonomous, and the latter are autonomous given nodes of the former.

3.1. The RAI Algorithm

Similarly to other algorithms of structure learning (Cheng et al., 1997; Cooper and Herskovits, 1992; Heckerman, 1995), the RAI algorithm[7] assumes that all the independences entailed from the given data can be encoded by a DAG. Similarly to other CB algorithms of structure learning (Cheng et al., 1997; Spirtes et al., 2000), the RAI algorithm assumes that the data sample size is large enough for reliable CI tests.

An iteration of the RAI algorithm starts with knowledge produced in the previous iteration and the current d-separation resolution, n. Previous knowledge includes $\mathcal{G}_{\text{start}}$, a structure having a d-separation resolution of $n-1$, and $\mathcal{G}_{\text{ex}}$, a set of structures each having possible exogenous causes to $\mathcal{G}_{\text{start}}$. Another input is the graph $\mathcal{G}_{\text{all}}$, which contains $\mathcal{G}_{\text{start}}$, $\mathcal{G}_{\text{ex}}$ and edges connecting them. Note that $\mathcal{G}_{\text{all}}$ may also contain other nodes and edges, which may not be required for the learning task (e.g., edges directed from nodes in $\mathcal{G}_{\text{start}}$ into nodes that are not in $\mathcal{G}_{\text{start}}$ or $\mathcal{G}_{\text{ex}}$), and these will be ignored by the RAI. In the first iteration, $n = 0$, $\mathcal{G}_{\text{ex}} = \emptyset$, $\mathcal{G}_{\text{start}}(\boldsymbol{V}, \boldsymbol{E})$ is the complete undirected graph and the d-separation resolution is not defined, since there are no pairs of d-separated nodes. Since $\mathcal{G}_{\text{ex}}$ is empty, $\mathcal{G}_{\text{all}} = \mathcal{G}_{\text{start}}$.

Given a structure $\mathcal{G}_{\text{start}}$ having d-separation resolution $n-1$, the RAI algorithm seeks independences between adjacent nodes conditioned on sets of size n and removes the

7. The RAI algorithm and a preliminary experimental evaluation of the algorithm were introduced in Yehezkel and Lerner (2005).

edges corresponding to these independences. The resulting structure has a d-separation resolution of n. After applying orientation rules so as to direct the remaining edges, a partial topological order is obtained in which parent nodes precede their descendants. Childless nodes have the lowest topological order. This order is partial, since not all the edges can be directed; thus, edges that cannot be directed connect nodes of equal topological order. Using this partial topological ordering, the algorithm decomposes the structure into ancestor and descendent autonomous sub-structures so as to reduce the complexity of the successive stages.

First, descendant sub-structures are established containing the lowest topological order nodes. A descendant sub-structure may be composed of a single childless node or several adjacent childless nodes. We will further refer to a single descendent sub-structure, although such a sub-structure may consist of several non-connected sub-structures. Second, all edges pointing towards nodes of the descendant sub-structure are temporarily removed (together with the descendant sub-structure itself), and the remaining clusters of connected nodes are identified as ancestor sub-structures. The descendent sub-structure is autonomous, given nodes of higher topological order composing the ancestor sub-structures. To consider smaller numbers of parents (and thereby smaller condition set sizes) when CI testing nodes of the descendant sub-structure, the algorithm first learns ancestor sub-structures, then the connections between ancestor and descendant sub-structures, and finally the descendant sub-structure itself. Each ancestor or descendent sub-structure is further learned by recursive calls to the algorithm. Figures 4, 5 and 6 show, respectively, the RAI algorithm, a manifesting example and the algorithm execution order for this example.

The RAI algorithm is composed of four stages (denoted in Figure 4 as Stages A, B, C and D) and an exit condition checked before the execution of any of the stages. The purpose of the exit condition is to assure that a CI test of a required order can indeed be performed, that is, the number of potential parents required to perform the test is adequate. The purpose of Stage A1 is to thin the link between $\mathcal{G}_{\text{ex}}$ and $\mathcal{G}_{\text{start}}$, the latter having d-separation resolution of $n-1$. This is achieved by removing edges corresponding to independences between nodes in $\mathcal{G}_{\text{ex}}$ and nodes in $\mathcal{G}_{\text{start}}$ conditioned on sets of size n of nodes that are either exogenous to, or within, $\mathcal{G}_{\text{start}}$. Similarly, in Stage B1, the algorithm tests for CI of order n between nodes in $\mathcal{G}_{\text{start}}$ given sets of size n of nodes that are either exogenous to, or within, $\mathcal{G}_{\text{start}}$, and removes edges corresponding to independences. The edges removed in Stages A1 and B1 could not have been removed in previous applications of these stages using condition sets of lower orders. When testing independence between X and Y, conditioned on the potential parents of node X, those nodes in the condition set that are exogenous to $\mathcal{G}_{\text{start}}$ are X's parents whereas those nodes that are in $\mathcal{G}_{\text{start}}$ are either its parents or adjacents.

In Stages A2 and B2, the algorithm directs every edge from the remaining edges that can be directed. In Stage B3, the algorithm groups in a descendant sub-structure all the nodes having the lowest topological order in the derived partially directed structure, and following the temporary removal of these nodes, it defines in Stage B4 separate ancestor sub-structures. Due to the topological order, every edge from a node X in an ancestor sub-structure to a node Z in the descendant sub-structure is directed as $X \rightarrow Z$. In addition, there is no edge connecting one ancestor sub-structure to another ancestor sub-structure.

Thus, every ancestor sub-structure contains all the potential parents of its nodes, that is, it is autonomous (or if some potential parents are exogenous, then the sub-structure is autonomous given the set of exogenous nodes). The descendant sub-structure is, by

Main function: $\mathcal{G}_{\text{out}} = \text{RAI}[n, \mathcal{G}_{\text{start}}(\boldsymbol{V}_{\text{start}}, \boldsymbol{E}_{\text{start}}), \mathcal{G}_{\text{ex}}(\boldsymbol{V}_{ex}, \boldsymbol{E}_{ex}), \mathcal{G}_{\text{all}}]$

Exit condition

If all nodes in $\mathcal{G}_{\text{start}}$ have fewer than $n+1$ potential parents, set $\mathcal{G}_{\text{out}} = \mathcal{G}_{\text{all}}$ and exit.

A. **Thinning the link between $\mathcal{G}_{\text{ex}}$ and $\mathcal{G}_{\text{start}}$ and directing $\mathcal{G}_{\text{start}}$**

1. For every node Y in $\mathcal{G}_{\text{start}}$ and its parent X in $\mathcal{G}_{\text{ex}}$, if $\exists \boldsymbol{S} \subset \{\boldsymbol{Pa}_p(Y, \mathcal{G}_{\text{start}}) \cup \boldsymbol{Pa}(Y, \mathcal{G}_{\text{ex}}) \backslash X\}$ and $|\boldsymbol{S}| = n$ such that $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$, then remove the edge between X and Y from $\mathcal{G}_{\text{all}}$.
2. Direct the edges in $\mathcal{G}_{\text{start}}$ using orientation rules.

B. **Thinning, directing and decomposing $\mathcal{G}_{\text{start}}$**

1. For every node Y and its potential parent X both in $\mathcal{G}_{\text{start}}$, if $\exists \boldsymbol{S} \subset \{\boldsymbol{Pa}(Y, \mathcal{G}_{\text{ex}}) \cup \boldsymbol{Pa}_p(Y, \mathcal{G}_{\text{start}}) \backslash X\}$ and $|\boldsymbol{S}| = n$ such that $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$, then remove the edge between X and Y from $\mathcal{G}_{\text{all}}$ and $\mathcal{G}_{\text{start}}$.
2. Direct the edges in $\mathcal{G}_{\text{start}}$ using orientation rules.
3. Group the nodes having the lowest topological order into a descendant sub-structure $\mathcal{G}_{\text{D}}$.
4. Remove $\mathcal{G}_{\text{D}}$ from $\mathcal{G}_{\text{start}}$ temporarily and define the resulting unconnected structures as ancestor sub-structures $\mathcal{G}_{\text{A}_1}, \ldots, \mathcal{G}_{\text{A}_k}$.

C. **Ancestor sub-structure decomposition**

For $i = 1$ to k, call $\text{RAI}[n+1, \mathcal{G}_{\text{A}_i}, \mathcal{G}_{\text{ex}}, \mathcal{G}_{\text{all}}]$.

D. **Descendant sub-structure decomposition**

1. Define $\mathcal{G}_{\text{ex}_\text{D}} = \{\mathcal{G}_{\text{A}_1}, \ldots, \mathcal{G}_{\text{A}_k}, \mathcal{G}_{\text{ex}}\}$ as the exogenous set to $\mathcal{G}_{\text{D}}$.
2. Call $\text{RAI}[n+1, \mathcal{G}_{\text{D}}, \mathcal{G}_{\text{ex}_\text{D}}, \mathcal{G}_{\text{all}}]$.
3. Set $\mathcal{G}_{\text{out}} = \mathcal{G}_{\text{all}}$ and exit.

Figure 4: The RAI algorithm.

definition, autonomous given nodes of ancestor sub-structures. Proposition 5 showed that we can identify all the conditional independences between nodes of an autonomous sub-structure. Hence, every ancestor and descendant sub-structure can be processed independently in Stages C and D, respectively, so as to identify conditional independences of increasing orders in each recursive call of the algorithm. Stage C is a recursive call for the RAI algorithm for learning each ancestor sub-structure with order $n+1$. Similarly, Stage D is a recursive call for the RAI algorithm for learning the descendant sub-structure with order $n+1$, while assuming that the ancestor sub-structures have been fully learned (having d-separation resolution of $n+1$).

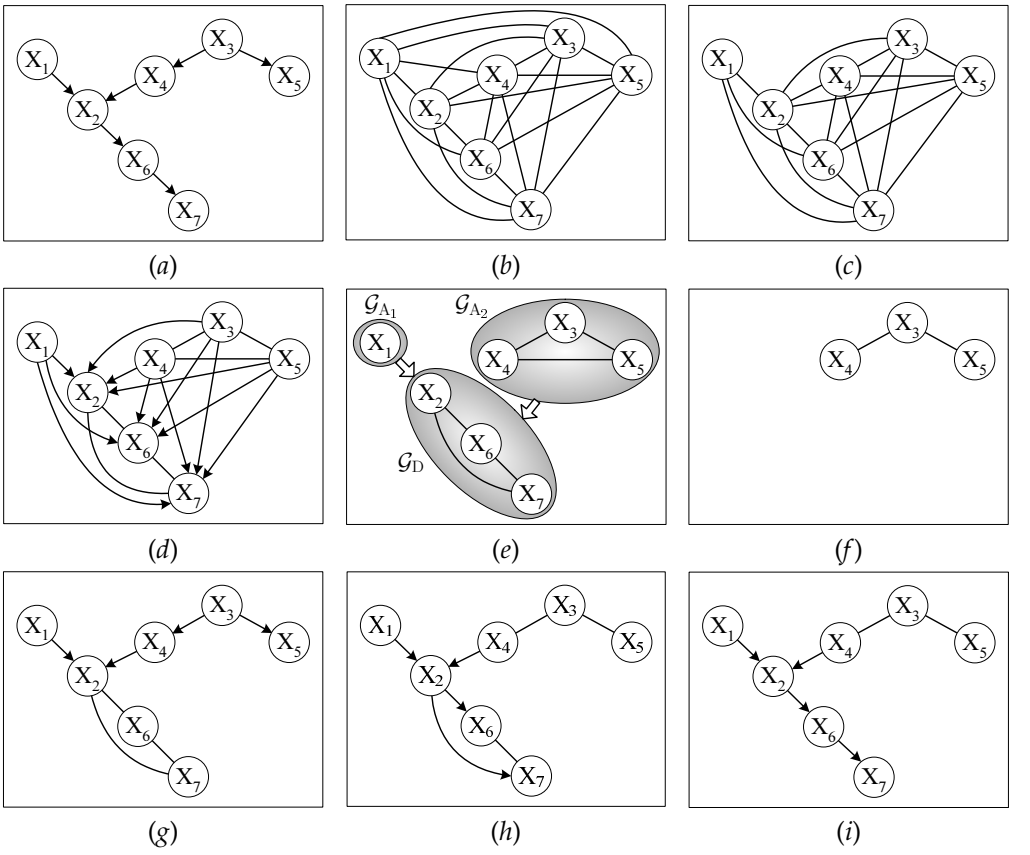


Figure 5: Learning an example structure. a) The true structure to learn, b) initial (complete) structure and structures learned by the RAI algorithm in Stages (see Figure 4) c) B1, d) B2, e) B3 and B4, f) C, g) D and A1, h) D and A2 and i) D, B1 and B2 (i.e., the resulting structure).

Figure 5 and Figure 6, respectively, show diagrammatically the stages in learning an example graph and the execution order of the algorithm for this example. Figure 5a shows the true structure that we wish to uncover. Initially, $\mathcal{G}_{\text{start}}$ is the complete undirected graph (Figure 5b), $n = 0$, $\mathcal{G}_{\text{ex}}$ is empty and $\mathcal{G}_{\text{all}} = \mathcal{G}_{\text{start}}$, so Stage A is skipped. In Stage B1, any pair of nodes in $\mathcal{G}_{\text{start}}$ is CI tested given an empty condition set

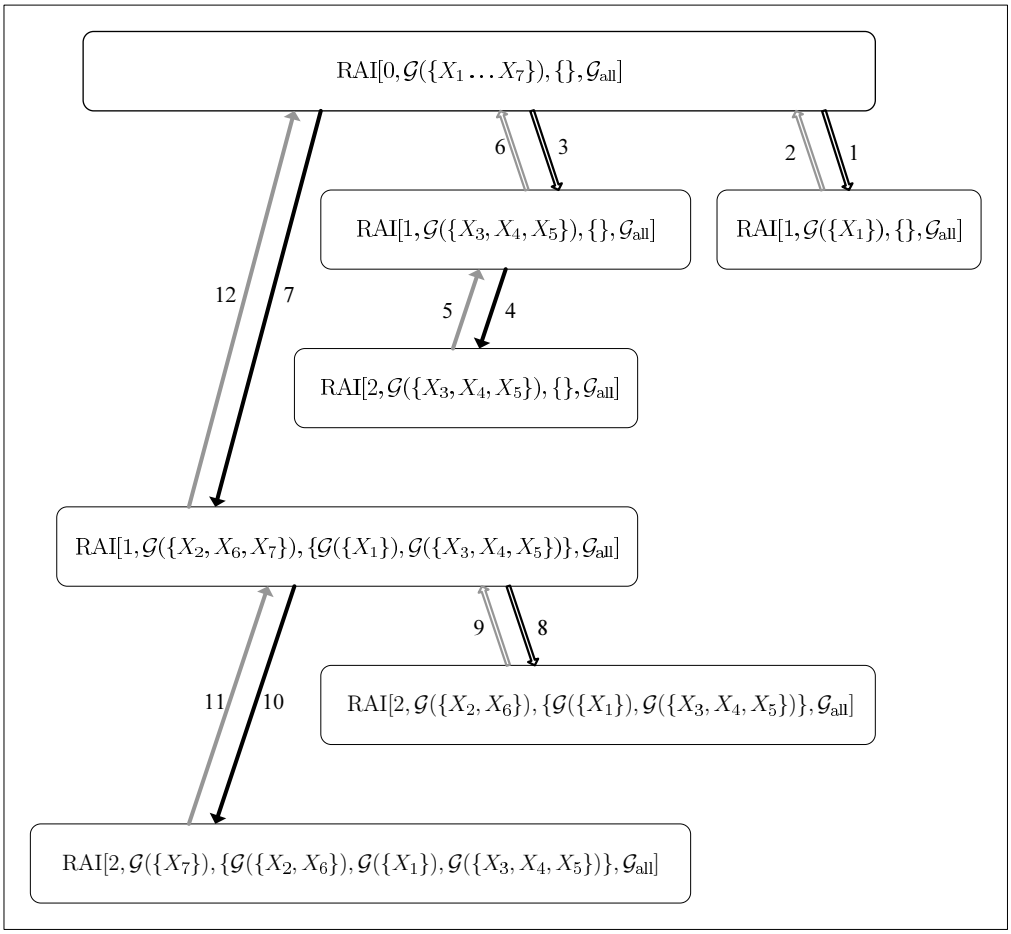


Figure 6: The execution order of the RAI algorithm for the example structure of Figure 5. Recursive calls of Stages C and D are marked with double and single arrows, respectively. The numbers annotating the arrows indicate the order of calls and returns of the algorithm.

(i.e., checking marginal independence), which yields the removal of the edges between node X_1 and nodes X_3, X_4 and X_5 (Figure 5c). The edge directions inferred in Stage B2 are shown in Figure 5d. The nodes having the lowest topological order (X_2, X_6, X_7) are grouped into a descendant sub-structure $\mathcal{G}_{\mathrm{D}}$ (Stage B3), while the remaining nodes form two unconnected ancestor sub-structures, $\mathcal{G}_{\mathrm{A}_1}$ and $\mathcal{G}_{\mathrm{A}_2}$ (Stage B4)(Figure 5e). Note that after decomposition, every edge between a node, X_i, in an ancestor sub-structure, and a node, X_j, in a descendant sub-structure is a directed edge $X_i \rightarrow X_j$. The set of all edges from an ancestor sub-structure to the descendant sub-structure is illustrated in Figure 5e by a wide arrow connecting the sub-structures. In Stage C, the algorithm is called recursively for each of the ancestor sub-structures with $n = 1$, $\mathcal{G}_{\text{start}} = \mathcal{G}_{\mathrm{A}_i}$ ($i = 1, 2$) and $\mathcal{G}_{\text{ex}} = \emptyset$. Since sub-structure $\mathcal{G}_{\mathrm{A}_1}$ contains a single node, the exit condition for this structure is satisfied. While calling $\mathcal{G}_{\text{start}} = \mathcal{G}_{\mathrm{A}_2}$, Stage A is skipped, and in Stage B1 the algorithm identifies that $X_4 \perp\!\!\!\perp X_5 \mid X_3$, thus removing the edge $X_4 - X_5$. No orientations are identified (e.g., X_3 cannot be a collider, since it separated X_4 and X_5), so the three nodes have equal topological order and they are grouped to form a descendant sub-structure. The recursive call for this sub-structure with $n = 2$ is returned immediately, since the exit condition is satisfied (Figure 5f). Moving to Stage D, the RAI is called with $n = 1$, $\mathcal{G}_{\text{start}} = \mathcal{G}_{\mathrm{D}}$ and $\mathcal{G}_{\text{ex}} = \{\mathcal{G}_{\mathrm{A}_1}, \mathcal{G}_{\mathrm{A}_2}\}$. Then, in Stage A1 relations $X_1 \perp\!\!\!\perp \{X_6, X_7\} \mid X_2$, $X_4 \perp\!\!\!\perp \{X_6, X_7\} \mid X_2$ and $\{X_3, X_5\} \perp\!\!\!\perp \{X_2, X_6, X_7\} \mid X_4$ are identified, and the corresponding edges are removed (Figure 5g). In Stage A2, X_6 and X_7 cannot collide at X_2 (since X_6 and X_7 are adjacent), and X_2 and X_6 (X_7) cannot collide at X_7 (X_6) (since X_2 and X_6 (X_7) are adjacent); hence, no additional V-structures are formed. Based on the inductive step and since X_1 is directed at X_2, X_2 should be directed at X_6 and at X_7. X_6 (X_7) cannot be directed at X_7 (X_6), because no new V-structures are allowed (Figure 5h). Stage B1 of the algorithm identifies the relation $X_2 \perp\!\!\!\perp X_7 \mid X_6$ and removes the edge $X_2 \rightarrow X_7$. In Stage B2, X_6 cannot be a collider of X_2 and X_7, since it has separated them. In the inductive step, X_6 is directed at X_7, $X_6 \rightarrow X_7$ (Figure 5i). In Stages B3 and B4, X_7 and $\{X_2, X_6\}$ are identified as a descendant sub-structure and an ancestor sub-structure, respectively. Further recursive calls (8 and 10 in Figure 6) are returned immediately, and the resulting partially directed structure (Figure 5i) represents a family of Markov equivalent structures (pattern) of the true structure (Figure 5a).

3.2. Minimality, Stability and Complexity

After describing the RAI algorithm (Section 3.1) and before proving its correctness (Section 3.3), we analyze in Section 3.2 three essential aspects of the algorithm — minimality, stability and complexity.

3.2.1. MINIMALITY

A structure recovered by the RAI algorithm in iteration m has a higher d-separation resolution and entails fewer dependences and thus is simpler and preferred[8] to a structure recovered in iteration $m - k$ where $0 < k \leq m$. By increasing the resolution, the RAI algorithm, similarly to the PC algorithm, moves from a complete undirected graph having maximal dependence relations between variables to structures having

8. We refer here to structures learned during algorithm execution and do not consider the empty graph that naturally has the lowest d-separation resolution (i.e., 0). This graph, having all nodes marginally independent of each other, will be found by the RAI algorithm immediately after the first iteration for graph resolution 0.

less (or equal) dependences than previous structures, ending in a structure having no edges between conditionally independent nodes, that is, a minimal structure.

3.2.2. STABILITY

Similarly to Spirtes et al. (2000), we use the notion of stability informally to measure the number of errors in the output of a stage of the algorithm due to errors in the input to this stage. Similarly to the PC algorithm, the main sources of errors of the RAI algorithm are CI-testing and the identification of V-structures. Removal of an edge due to an erroneous CI test may lead to failure in correctly removing other edges, which are not in the true graph and also cause to orientation errors. Failure to remove an edge due to an erroneous CI test may prevent, or wrongly cause, orientation of edges. Missing or wrongly identifying a V-structure affect the orientation of other edges in the graph during the inductive stage and subsequent stages.

Many CI test errors (i.e., deciding that (in)dependence exists where it does not) in CB algorithms are the result of unnecessary large condition sets given a limited database size (Spirtes et al., 2000). Large condition sets are more likely to be inaccurate, since they are more likely to include unnecessary and erroneous nodes (erroneous due to errors in earlier stages of the algorithm). These sets may also cause poorer estimation of the criterion that measures dependence (e.g., CMI or χ^2) due to the curse-of-dimensionality, as typically there are only too few instances representing some of the combinations of node states. Either way, these condition sets are responsible for many wrong decisions about whether dependence between two nodes exists or not. Consequently, these errors cause structural inaccuracies and hence also poor inference ability.

Although CI-testing in the PC algorithm is more stable than V-structure identification (Spirtes et al., 2000), it is difficult to say whether this is also the case in the RAI algorithm. Being recursive, the RAI algorithm might be more unstable. However, CI test errors are practically less likely to occur, since by alternating between CI testing and edge direction the algorithm uses knowledge about parent-child relations before CI testing of higher orders. This knowledge permits avoiding some of the tests and decreases the size of conditions sets of some other tests (see Lemma 6). In addition, graph decomposition promotes decisions about well-founded orders of node presentation for subsequent CI tests, contrary to the common arbitrary order of presentation (see, e.g., the PC algorithm). Both mechanisms enhance stability and provide some means of error correction, as will be demonstrated shortly.

Let us now extensively describe examples that support our claim regarding the enhanced stability of the RAI algorithm. Suppose that following CI tests of some order both the PC and RAI algorithms identify a triplet of nodes in which two non-adjacent nodes, X and Y, are adjacent to a third node, Z, that is, $X - Z - Y$. In the immediate edge direction stage, the RAI algorithm identifies this triplet as a V-structure, $X \rightarrow Z \leftarrow Y$. Now, suppose that due to an unreliable CI test of a higher order the PC algorithm removes $X - Z$ and the RAI algorithm removes $X \rightarrow Z$. Eventually, both algorithms fail to identify the V-structure, but the RAI algorithm has an advantage over the PC algorithm in that the other arm of the V-structure is directed, $Z \leftarrow Y$. This contributes to the possibility to direct further edges during the inductive stage and subsequent recursive calls for the algorithm. The directed arm would also contribute to fewer CI tests and tests with smaller condition sets during CI testing with higher orders (e.g., if we later have to test independence between Y and another node, then we know that Z should not be included in the condition set, even though it is adjacent to Y). In addition, the direction of this edge also contributes to enhanced inference capability.

Now, suppose another example in which after removing all edges due to reliable CI tests using condition set sizes lower than or equal to n, the algorithm identifies the V-structure $X \rightarrow Z \leftarrow Y$ (Figure 7a). However, let assume that one of the V-structure arms, say $X \rightarrow Z$, is correctly removed on a subsequent iteration using a larger condition set size (say $n+1$ without limiting the generality). We may be concerned that assuming a V-structure for the lower graph resolution, the RAI algorithm wrongly directs the second arm $Z - Y$ as $Z \leftarrow Y$. However, we demonstrate that the edge direction $Z \leftarrow Y$ remains valid even if there should be no edge $X - Z$ in the true graph. Suppose that $X \rightarrow Z$ was correctly removed conditioned on variable W, which is independent of Y given any condition set with a size smaller than or equal to n. Then, the possible underlying graphs are shown in Figures 7b-7d. The graph in Figure 7d is not possible, since it yields that X and Y are dependent given all condition sets of sizes smaller than or equal to n. In Figure 7b and Figure 7c, Z is a collider between W and Y, and thus the edge direction $Z \leftarrow Y$ remains valid. A different graph, $X \rightarrow W \leftarrow Z - Y$ (i.e., W is a collider), is not possible, since it means that $X \perp\!\!\!\perp Z \,|\, \boldsymbol{S}$, $|\boldsymbol{S}| \leq n$, $W \notin \boldsymbol{S}$ and then $X - Z$ should have been removed in a previous order (using condition set size of n or lower) and $X \rightarrow Z \leftarrow Y$ should not have been identified in the first place. Now, suppose that W and Y are dependant. In this case, the possible graphs are those shown in Figures 7e-7h. Similarly to the case in which W and Y are independent, W cannot be a collider of X and Z ($X \rightarrow W \leftarrow Z$) in this case as well. The graphs shown in Figures 7e-7g cannot be the underlying graphs since they entail dependency between X and Y given a condition set of size lower than or equal to n. The graph shown in Figure 7h exemplifies a V-structure $X \rightarrow W \leftarrow Y$. Since we assume that X and Z are independent given W (and thus $X - Z$ was removed), a V-structure $X \rightarrow W \leftarrow Z$ is not allowed. Since the edge $X \rightarrow W$ is already directed, the edge between W and Z must be directed as $W \rightarrow Z$. In this case, to avoid the cycle $Y \rightarrow W \rightarrow Z \rightarrow Y$, the edge between Y and Z must be directed as in the true graph, that is, $Y \rightarrow Z$.

Finally for the stability subsection, we note that the contribution of graph decomposition to structure learning using the RAI algorithm is threefold. First is the identification in early stages, using low-order, reliable CI tests, of the graph hierarchy, exemplifying the backbone of causal relations in the graph. For example, Figure 5e shows that learning our example graph (Figure 5a) from the complete graph (Figure 5b) demonstrates, immediately after the first iteration, that the graph is composed of three sub-structures — $\{X_1\}$, $\{X_2, X_6, X_7\}$ and $\{X_3, X_4, X_5\}$, where $\{X_1\} \rightarrow \{X_2, X_6, X_7\}$ and $\{X_3, X_4, X_5\} \rightarrow \{X_2, X_6, X_7\}$. This rough (low-resolution) partition of the graph is helpful in visualizing the problem and representing the current knowledge from the outset and along the learning. The second contribution of graph decomposition is the possibility to implement learning using a parallel processor for each sub-structure independently. This advantage may be further extended in the recursive calls for the algorithm.

Third is the contribution of graph decomposition to improved performance. Aiming at a low number of CI tests, decomposition provides a sound guideline for deciding on an educated order in which the edges should be CI tested. Based on this order, some tests can be considered redundant and thus be avoided. Several methods for selecting the right order for the PC algorithm were presented in Spirtes et al. (2000), but these methods are heuristic. Decomposition into ancestor and descendent sub-structures is followed by three levels of learning (Figure 4), that is, removing and directing edges 1) of ancestor sub-structures, 2) between ancestor and descendent sub-structures, and 3) of the descendent sub-structure. The second level has the greatest influence on

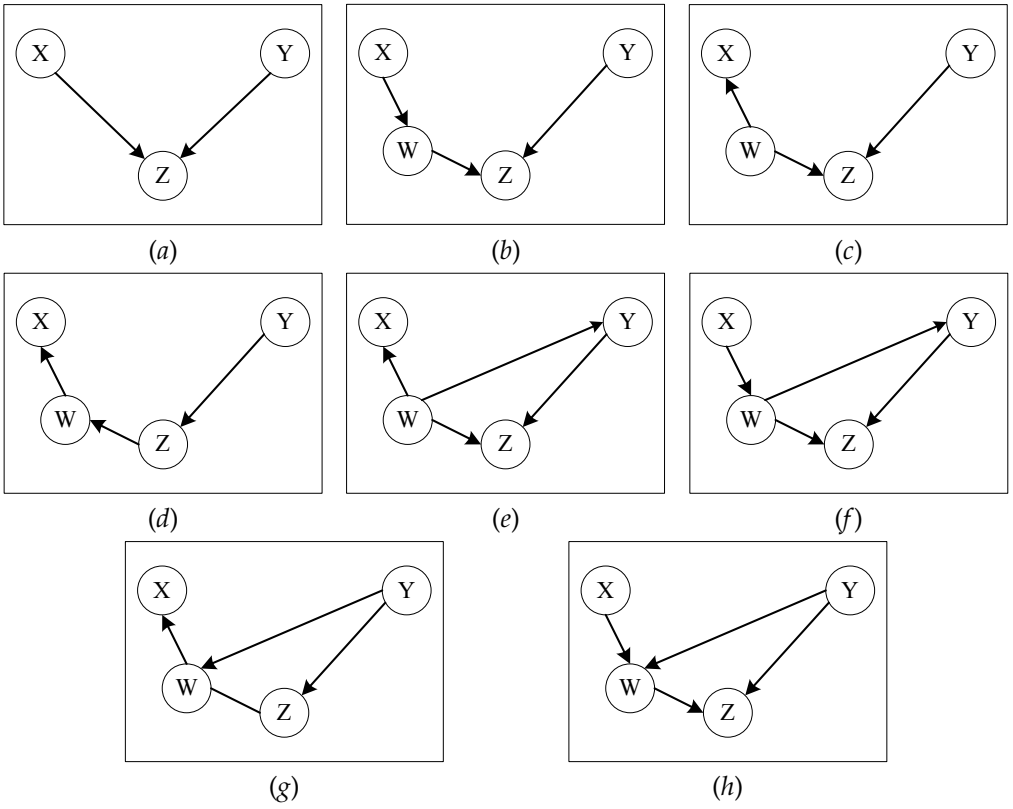


Figure 7: Graphs used to exemplify the stability of the RAI algorithm (see text).

further learning. The removal of edges between ancestor and descendent sub-structures and the sequential direction of edges in the descendant sub-structure assure that, first, fewer potential parents are considered, while learning the descendent sub-structure and second, more edges can be directed in this latter sub-structure. Moreover, these directed edges and the derived parent-child relations prevent an arbitrary selection order of nodes for CI testing and thereby enable employing smaller and more accurate condition sets. Take, for example, CI testing for the redundant edge between X_2 and X_7 in our example graph (Figure 5i) if the RAI algorithm did not use decomposition. Graph decomposition for $n = 0$ (Figure 5e) enables the identification of two ancestor sub-structures, $\mathcal{G}_{A_1}$ and $\mathcal{G}_{A_2}$, as well as a descendent sub-structure $\mathcal{G}_D$ that are each learned recursively. During Stage D (Figure 4) and while thinning the links between the ancestor sub-structures and $\mathcal{G}_D$ (in Stage A1 of the recursion for $n = 1$), we identify the relations $X_1 \perp\!\!\!\perp \{X_6, X_7\} \mid X_2$, $X_4 \perp\!\!\!\perp \{X_6, X_7\} \mid X_2$ and $\{X_3, X_5\} \perp\!\!\!\perp \{X_2, X_6, X_7\} \mid X_4$ and remove the 10 corresponding edges (Figure 5g). The decision to test and remove these edges first was enabled by the decomposition of the graph to $\mathcal{G}_{A_1}$, $\mathcal{G}_{A_2}$ and $\mathcal{G}_D$. In Stage A2 (Figure 5h), we direct the edge $X_2 \to X_6$ (as $X_1 \perp\!\!\!\perp X_6 \mid X_2$ and thus X_2 cannot be a collider between X_1 and X_6) and edge $X_2 \to X_7$ (as $X_1 \perp\!\!\!\perp X_7 \mid X_2$ and thus X_2 cannot be a collider between X_1 and X_7), and in Stage B (Figure 5i) we direct the edge $X_6 \to X_7$. The direction of these edges could not be assured without removing first the above edges, since the (redundant) edges pointing onto X_6 and X_7 would have allowed wrong edge direction, that is, $X_6 \to X_2$ and $X_7 \to X_2$. If we had been using the RAI algorithm with no decomposition (Figure 5d) (or the PC algorithm) and had decided to check the independence between X_2 and X_7, first, we would have had to consider condition sets containing the nodes X_1, X_3, X_4, X_5 or X_6 (up to 10 CI tests whether we start from X_2 or X_7). Instead, we perform in Stage B1 only one test, $X_2 \perp\!\!\!\perp X_7 \mid X_6$. These benefits are the result of graph decomposition.

3.2.3. COMPLEXITY

CI tests are the major contributors to the (run-time) complexity of CB algorithms (Cheng and Greiner, 1999). In the worst case, the RAI algorithm will neither direct any edges nor decompose the structure and will thus identify the entire structure as a descendant sub-structure, calling Stages D and B1 iteratively while skipping all other stages. Then, the execution of the algorithm will be similar to that of the PC algorithm, and thus the complexity will be bounded by that of the PC algorithm. Given the maximal number of possible parents k and the number of nodes n, the number of CI tests is bounded by (Spirtes et al., 2000)

$$2\binom{n}{2} \cdot \sum_{i=0}^{k} \binom{n-1}{i} \leq \frac{n^2(n-1)^{k-1}}{(k-1)!},$$

which leads to complexity of $O(n^k)$.

This bound is loose even in the worst case (Spirtes et al., 2000) especially in real-world applications requiring graphs having V-structures. This means that in most cases some edges are directed and the structure is decomposed; hence, the number of CI tests is much smaller than that of the worst case. For example, by decomposing our example graph (Figure 5) into descendent and ancestor sub-structures in the first application of Stage B4 (Figure 5e), we avoid checking $X_6 \perp\!\!\!\perp X_7 \mid \{X_1, X_3, X_4, X_5\}$. This is because $\{X_1, X_3, X_4, X_5\}$ are neither X_6's nor X_7's parents and thus are not included in the (autonomous) descendent sub-structure. By checking only $X_6 \perp\!\!\!\perp X_7 \mid \{X_2\}$, the

RAI algorithm saves CI tests that are performed by the PC algorithm. We will further elaborate on the RAI algorithm complexity in our forthcoming study.

3.3. Proof of Correctness

We prove the correctness of the RAI algorithm using Proposition 7. We show that only conditional independences (of all orders) entailed by the true underlying graph are identified by the RAI algorithm and that all V-structures are correctly identified. We then note on the correctness of edge direction.

Proposition 7 *If the input data to the RAI algorithm are faithful to a DAG, $\mathcal{G}_{\text{true}}$, having any d-separation resolution, then the algorithm yields the correct pattern for $\mathcal{G}_{\text{true}}$.*

Proof We use mathematical induction to prove the proposition, where in each induction step, m, we prove that the RAI algorithm finds (a) all conditional independences of order m and lower, (b) no false conditional independences, (c) only correct V-structures and (d) all V-structures, that is, no V-structures are missing.

Base step ($m = 0$): If the input data to the RAI algorithm was generated from a distribution faithful to a DAG, $\mathcal{G}_{\text{true}}$, having d-separation resolution 0, then the algorithm yields the correct pattern for $\mathcal{G}_{\text{true}}$.

Given that the true underlying DAG has a d-separation resolution of 0, the data entail only marginal independences. In the beginning of learning, $\mathcal{G}_{\text{start}}$ is a complete graph and $m = 0$. Since there are no exogenous causes, Stage A is skipped. In Stage B, the algorithm tests for independence between every pair of nodes with an empty condition set, that is, $X \perp\!\!\!\perp Y \,|\, \emptyset$ (marginal independence), removes the redundant edges and directs the remaining edges as possible. In the resulting structure, all the edges between independent nodes have been removed and no false conditional independences are entailed. Thus, all the identified V-structures are correct, as discussed in Section 3.2.2 on stability, and there are no missing V-structures, since the RAI algorithm has tested independence for all pair of nodes (edges). At the end of Stage B2 (edge direction), the resulting structure and $\mathcal{G}_{\text{true}}$ have the same set of V-structures and the same set of edges. Thus, the correct pattern for $\mathcal{G}_{\text{true}}$ is identified. Since the data entail only independences of zero order, further recursive calls with $m \geq 1$ will not find independences with condition sets of size m, and thus no edges will be removed, leaving the graph unchanged.

Inductive step ($m + 1$): Suppose that at induction step m, the RAI algorithm discovers all conditional independences of order m and lower, no false conditional independences are entailed, all V-structures are correct, and no V-structures are missing. Then, if the input data to the RAI algorithm was generated from a distribution faithful to a DAG, $\mathcal{G}_{\text{true}}$, having d-separation resolution $m + 1$, then the RAI algorithm would yield the correct pattern for that graph.

In step m, the RAI algorithm discovers all conditional independences of order m and lower. Given input data faithful to a DAG, $\mathcal{G}_{\text{true}}$, having d-separation resolution $m + 1$, there exists at least one pair of nodes, say $\{X, Y\}$, in the true graph, that has a d-separation resolution of $m + 1$.[9] Since the RAI, by the recursive call $m + 1$ (i.e., calling RAI$[m + 1, \mathcal{G}_{\text{start}}, \mathcal{G}_{\text{ex}}, \mathcal{G}_{\text{all}}]$), has identified only conditional independences of order m

9. If the d-separation resolution of $\{X, Y\}$ is $m' > m + 1$, then the RAI algorithm will not modify the graph until step m'.

and lower, an edge, $E_{XY} = (X - Y)$, exists in the input graph, $\mathcal{G}_{\text{start}}$. The smallest condition set required to identify the independence between X and Y is $\boldsymbol{S}_{XY}$ ($X \perp\!\!\!\perp Y \mid \boldsymbol{S}_{XY}$), such that $|\boldsymbol{S}_{XY}| \geq m+1$. Thus, $|\boldsymbol{Pa}_p(X) \backslash Y| \geq m+1$ or $|\boldsymbol{Pa}_p(Y) \backslash X| \geq m+1$, meaning that either node X or node Y has at least $m+2$ potential parents. Such an edge exists in at least one of the autonomous sub-structures decomposed from the graph yielded at the end of iteration m. When calling, in Stage C or Stage D, the algorithm recursively for this sub-structure with $m' = m+1$, the exit condition is not satisfied because either node X or node Y has at least $m'+1$ parents. Since Step m assured that the sub-structure is autonomous, it contains all the necessary node parents. Note that decomposition into ancestor, $\mathcal{G}_{\text{A}}$, and descendant, $\mathcal{G}_{\text{D}}$, sub-structures occurs after identification of all nodes having the lowest topological order, such that every edge from a node X in $\mathcal{G}_{\text{A}}$ to a node Y in $\mathcal{G}_{\text{D}}$ is directed, $X \rightarrow Y$. In the case that the sub-structure is an ancestor sub-structure, $\boldsymbol{S}_{XY}$ contains nodes of the sub-structure and its exogenous causes. In the case that the sub-structure is a descendant sub-structure, $\boldsymbol{S}_{XY}$ contains nodes from the ancestor sub-structures and the descendant sub-structure. Therefore, based on Proposition 5, the RAI algorithm tests all edges using condition sets of sizes m' and removes E_{XY} (and all similar edges) in either Stage A or Stage B, yielding a structure with d-separation resolution of m' and thereby yields the correct pattern for the true underlying graph of d-separation resolution $m+1$.

Spirtes (2001)—when introducing the anytime fast casual inference (AFCI) algorithm—proved the correctness of edge direction of AFCI. The AFCI algorithm can be interrupted at any stage (resolution), and the resultant graph at this stage is correct with probability one in the large sample limit, although possibly less informative[10] than if had been allowed to continue uninterrupted.[11] Recall that interrupting learning means that we avoid CI tests of higher orders. This renders the resultant graph more reliable. We use this proof here for proving the correctness of edge direction in the RAI algorithm. Completing CI testing with a specific graph resolution n in the RAI algorithm and interrupting the AFCI at any stage of CI testing are analogous. Furthermore, Spirtes (2001) proves that interrupting the algorithm at any stage is also possible during edge direction, that is, once an edge is directed, the algorithm never changes that direction. In Section 3.2.2, we showed that even if a directed edge of a V-structure is removed, the direction of the remaining edge is still correct. Since directing edges by the AFCI algorithm after interruption yields a correct (although less informative) graph (Spirtes, 2001), also the direction of edges by the RAI algorithm yields a correct graph. Having (real) parents in a condition set used for CI testing, instead of potential parents, which are the result of edge direction for resolutions lower than n, is a virtue, as was confirmed in Section 3.1. All that is required that all parents, either real or potential, be included within the corresponding condition set, and this is indeed guaranteed by the autonomy of each sub-structure, as was proved above. ■

10. Less informative in the sense that it answers "can't tell" for a larger number of questions; that is, identifying, for example, "$\circ$" edge endpoint (placing no restriction on the relation between the pair of nodes making the edge) instead of "$\rightarrow$" endpoint.
11. The AFCI algorithm is also correct if hidden and selection variables exist. A selection variable models the possibility of an observable variable having some missing data. We focus here on the case where neither hidden nor selection variables exist.

4. Experiments and Results

We compare the RAI algorithm with other state-of-the-art algorithms with respect to structural correctness, computational complexity, run-time and classification accuracy when the learned structure is used in classification. The algorithms learned structures from databases representing synthetic problems, real decision support systems and natural classification problems. We present the experimental evaluation in four sections. In Section 4.1, the complexity of the RAI algorithm is measured by the number of CI tests required for learning synthetically generated structures in comparison to the complexity of the PC algorithm (Spirtes et al., 2000).

The order of presentation of nodes is not an input to the PC algorithm. Nevertheless, CI testing of orders higher than 0, and therefore also edge directing, which depends on CI testing, may be sensitive to that order. This may cause learning different graphs whenever the order is changed. Dash and Druzdzel (1999) turned this vice of the PC algorithm into a virtue by employing the partially directed graphs formed by using different orderings for the PC algorithm as the search space from which the structure having the highest value of the K2 metric (Cooper and Herskovits, 1992) is selected. For the RAI algorithm, sensitivity to the order of presentation of nodes is expected to be reduced compared to the PC algorithm, since the RAI algorithm, due to edge direction and graph decomposition, decides on the order of performing most of the CI tests and does not use an arbitrary order (Section 3.2.2). Nevertheless, to account for the possible sensitivity of the RAI and PC algorithms to this order, we preliminarily employed 100 different permutations[12] of the order for each of ten Alarm network (Beinlich et al., 1989) databases. Since the results of these experiments had showed that the difference in performance for different permutations is slight, we further limited the experiments with the PC and RAI algorithms to a single permutation.

In Section 4.2, we present our methodology of selecting a threshold for RAI CI testing. We propose selecting a threshold for which the learned structure has a maximum of a likelihood-based score value.

In Section 4.3, we use the Alarm network (Beinlich et al., 1989), which is a widely accepted benchmark for structure learning, to evaluate the structural correctness of graphs learned by the RAI algorithm. The correctness of the structure recovered by RAI is compared to those of structures learned using other algorithms — PC, TPDA (Cheng et al., 1997), GES (Chickering, 2002; Meek, 1997), SC (Friedman et al., 1999) and MMHC (Tsamardinos et al., 2006a). The PC and TPDA algorithms are the most popular CB algorithms (Cheng et al., 2002; Kennett et al., 2001; Marengoni et al., 1999; Spirtes et al., 2000); GES and SC are state-of-the-art S&S algorithms (Tsamardinos et al., 2006a); and MMHC is a hybrid algorithm that has recently been developed and showed superiority, with respect to different criteria, over all the (non-RAI) algorithms examined here (Tsamardinos et al., 2006a). In addition to correctness, the complexity of the RAI algorithm, as measured through the enumeration of CI tests and log operations, is compared to those of the other CB algorithms (PC and TPDA) for the Alarm network.

In Section 4.4, we extend the examination of RAI in structure learning to known networks other than the Alarm. Although the Alarm is a popular benchmark network, many algorithms perform well for this network. Hence, it is important to examine RAI performance on other networks for which the true graph is known. In the comparison of RAI to other algorithms, we included all the algorithms of Section 4.3, as well as the

12. Dash and Druzdzel (1999) examined the relationships between the number of order permutations and the numbers of variables and instances. We fixed the number of order permutations at 100.

Optimal Reinsertion (OR) (Moore and Wong, 2003) algorithm and a greedy hill-climbing search algorithm with a Tabu list (GS) (Friedman et al., 1999). We compared algorithm performances with respect to structural correctness, run-time, number of statistical calls and the combination of correctness and run-time.

In Section 4.5, the complexity and run-time of the RAI algorithm are compared to those of the PC algorithm using nineteen natural databases. In addition, the classification accuracy of the RAI algorithm for these databases is compared to those of the PC, TPDA, GES, MMHC, SC and naive Bayesian classifier (NBC) algorithms. No structure learning is required for NBC and all the domain variables are used. This classifier is included in the study as a reference to a simple, yet accurate, classifier. Because we are interested in this section in classification, and a likelihood-based score does not reflect the importance of the class variable in structures used for classification (Friedman et al., 1997; Kontkanen et al., 1999; Grossman and Domingos, 2004; Yang and Chang, 2002), we prefer here the classification accuracy score in evaluating structure performance.

In the implementations of all sections, except Section 4.4, we were aided by the Bayes net toolbox (BNT) (Murphy, 2001), BNT structure learning package (Leray and François, 2004) and PowerConstructor software (Cheng, 1998) and evaluated all algorithms ourselves. In Section 4.4, we downloaded and used the results reported in Tsamardinos et al. (2006a) for the non-RAI algorithms and used the Causal Explorer algorithm library (Aliferis et al., 2003) (`http://www.dsl-lab.org/causal_explorer/index.html`). The Causal Explorer algorithm library makes use of methods and values of parameters for each algorithm as suggested by the authors of each algorithm (Tsamardinos et al., 2006a). For example, BDeu score (Heckerman et al., 1995) with equivalent sample size 10 for GS, GES, OR and MMHC; χ^2 p-values at the standard 5% for the MMHC's and PC's statistical thresholds; threshold of 1% for the TPDA mutual information test; the Bayesian scoring heuristic, equivalent sample size of 10 and maximum allowed sizes for the candidate parent set of 5 and 10 for SC; and maximum number of parents allowed of 5, 10 and 20 and maximum allowed run time, which is one and two times the time used by MMHC on the corresponding data set, for OR. The only parameter that requires optimization in the RAI algorithm (similar to the other CB algorithms - PC and TPDA) is the CI testing threshold. We use no prior knowledge to find this threshold but a training set for each database (see Section 4.2 for details). Note, however that we do not account for the time required for selecting the threshold when reporting the execution time.

4.1. Experimentation with Synthetic Data

The complexity of the RAI algorithm was evaluated in comparison to that of the PC algorithm by the number of CI tests required to learn synthetically generated structures. Since the true graph is known for these structures, we could assume that all CI tests were correct and compare the numbers of CI tests required by the algorithms to learn the true independence relationships. In one experiment, all 29,281 possible structures having 5 nodes were learned using the PC and RAI algorithms. The average number of CI tests employed by each algorithm is shown in Figure 8a for increasing orders (condition set sizes). Figure 8b depicts the average percentages of CI tests saved by the RAI algorithm compared to the PC algorithm for increasing orders. These percentages were calculated for each graph independently and then averaged. It is seen that the advantage of the RAI algorithm over the PC algorithm is more prominent for high orders.

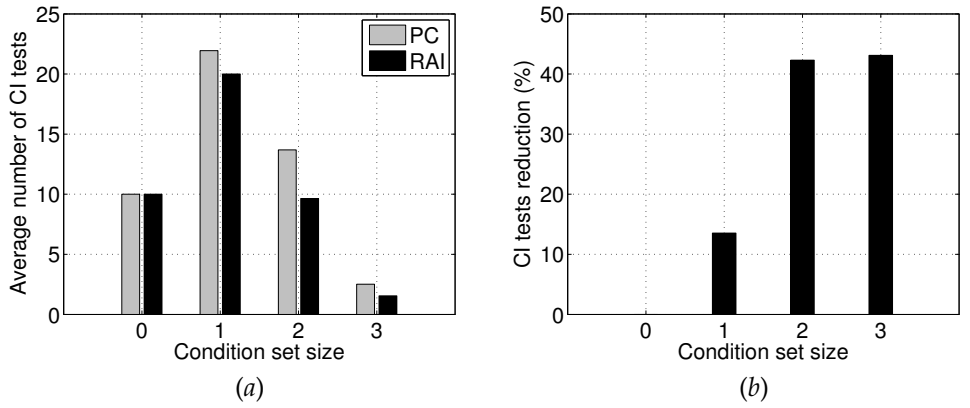


Figure 8: Measured for increasing orders, the (a) average number of CI tests required by the RAI and PC algorithms for learning all possible structures having five nodes and (b) average over all structures of the reduction percentage in CI tests achieved by the RAI algorithm compared to the PC algorithm.

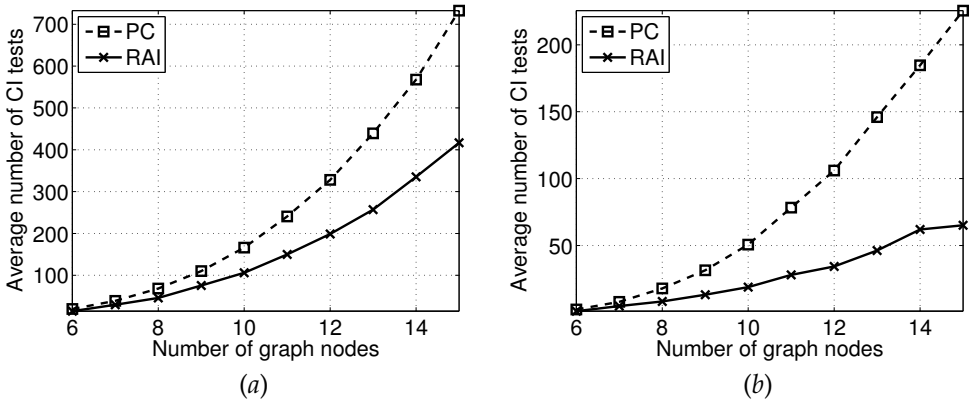


Figure 9: Average number of CI tests required by the PC and RAI algorithms for increasing graph sizes and orders of (a) 3 and (b) 4.

In another experiment, we learned graphs of sizes (numbers of nodes) between 6 and 15. We selected from a large number of randomly generated graphs 3,000 graphs that were restricted by a maximal fan-in value of 3; that is, every node in such a graph has 3 parents at most and at least one node in the graph has 3 parents. This renders a practical learning task. Thus, the structures can theoretically be learned by employing CI tests of order 3 and below and should not use tests of orders higher than 3. In such a case, the most demanding test, having the highest impact on computational time, is of order 3. Figure 9a shows the average numbers of CI tests performed for this order by the PC and RAI algorithms for graphs with increasing sizes. Moreover, because the

maximal fan-in is 3, all CI tests of order 4 are a priori redundant, so we can further check how well each algorithm avoids these unnecessary tests. Figure 9b depicts the average numbers of CI tests performed by the two algorithms for order 4 and graphs with increasing sizes. Both Figure 9a and Figure 9b show that the number of CI tests employed by the RAI algorithm increases more slowly with the graph size compared to that of the PC algorithm and that this advantage is much more significant for the redundant (and more costly) CI tests of order 4.

We further expanded the examination of the algorithms in CI testing for different graph sizes and CI test orders. Figure 10 shows the average number and percentage of CI tests saved using the RAI algorithm compared to the PC algorithm for different condition set sizes and graph sizes. The number of CI tests having an empty condition set employed by each of the algorithms is equal and is therefore omitted from the comparison. The figure shows that the percentage of CI tests saved using the RAI algorithm increases with both graph and condition set sizes. For example, the saving in CI tests when using the RAI algorithm instead of the PC algorithm for learning a graph having 15 nodes and using condition sets of size 4 is above 70% (Figure 10b). In Section 4.4, we will demonstrate the RAI quality of requiring relatively fewer tests of high orders than of low orders for graphs of larger sizes for real, rather than synthetic, data.

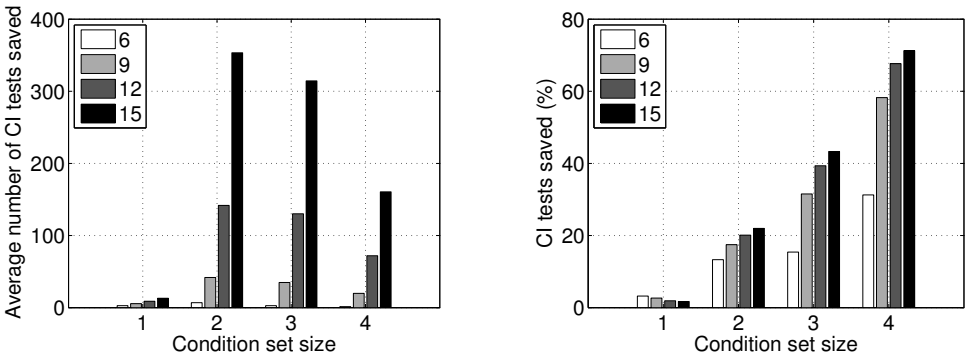


Figure 10: (a) Average number and (b) percentage of CI tests saved by using the RAI algorithm compared to the PC algorithm for graph sizes of 6, 9, 12 or 15 (gray shades) and orders between 1 and 4.

4.2. Selecting the Threshold for RAI CI Testing

CI testing for the RAI algorithm can be based on the χ^2 test as for the PC algorithm or the conditional mutual information (CMI) as for the TPDA algorithm. The CMI between nodes X and Y conditioned on a set of nodes $\boldsymbol{Z}$ (i.e., the condition set), is:

$$\mathrm{CMI}(X,Y|\boldsymbol{Z}) = \sum_{i=1}^{N_X}\sum_{j=1}^{N_Y}\sum_{k=1}^{N_Z}\left[P(x_i,y_j,z_k)\cdot\log\frac{P(x_i,y_j|z_k)}{P(x_i|z_k)\cdot P(y_j|z_k)}\right], \qquad (2)$$

where x_i and y_j represent, respectively, states of X and Y, z_k represents a combination of states of all variables in $\mathbf{Z}$, and N_X, N_Y and N_Z are the numbers of states of X, Y and $\mathbf{Z}$, respectively.

In both CI testing methods, the value of interest (either χ^2 or CMI) is compared to a threshold. For example, CMI values that are higher or lower than the threshold indicate, respectively, conditional dependence or independence between X and Y given $\mathbf{Z}$. However, the optimal threshold is unknown beforehand. Moreover, the optimal threshold is problem and data-driven, that is, it depends, on the one hand, on the database and its size and, on the other hand, on the variables and the numbers of their states. Thus, it is not possible to set a "default" threshold value that will accurately determine conditional (in)dependence while using any database or problem.

To find an optimal threshold for a database, we propose to score structures learned using different thresholds by a likelihood-based criterion evaluated using the training (actually validation) set and to select the threshold leading to the structure achieving the highest score. Such a score may be BDeu (Heckerman et al., 1995), although other scores (Heckerman et al., 1995) may also be appropriate. Note that BDeu scores equally statistically indistinguishable structures. Figure 11 shows BDeu values for structures learned by RAI for the Alarm network using different CMI threshold values. The maximum BDeu value was achieved at a threshold value of 4e-3 that was selected as the threshold for RAI CI testing for the Alarm network.

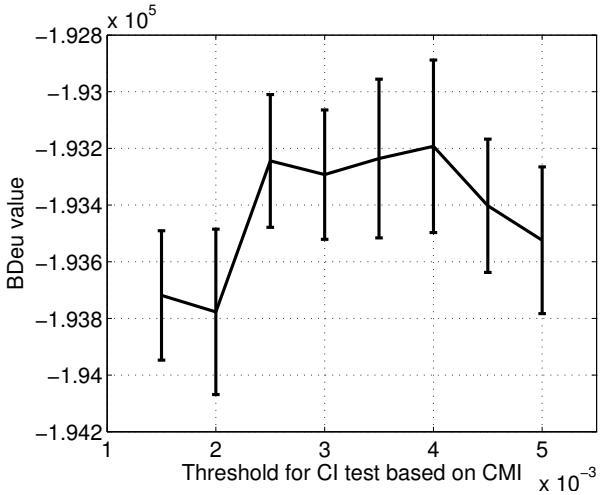


Figure 11: BDeu values averaged over ten validation sets consisting of 10,000 samples each drawn from the Alarm network for increasing CMI thresholds used in CI testing for the RAI algorithm.

To assess the threshold selected using the suggested method, we employed the Alarm network and computed the errors between structures learned using different thresholds and the pattern that corresponds to the true known graph. Following Spirtes et al. (2000) and Tsamardinos et al. (2006a), we define five types of structural errors to evaluate structural correctness. An extra edge (commission; EE) error is due to an edge learned by the algorithm although it does not exist in the true graph. A missing edge (omission; ME) error is due to an edge missed by the algorithm although exists

in the true graph. An extra direction (ED) error is due to edge direction that appears in the learned graph but not in the true graph, whereas a missing direction (MD) error is due to edge direction that appears in the true graph but not in the learned graph. Finally, a reversed direction (RD) error is due to edge direction in the learned graph that is opposite to the edge direction in the true graph.

Figure 12a shows the sensitivity of the five structural errors to the CMI threshold. Each point on the graph is the average error over ten validation databases containing 10,000 randomly sampled instances each. Figure 12a demonstrates that the MD, RD and ED errors are relatively constant in the examined range of thresholds and the ME error increases monotonically. The EE error is the highest error among the five error types, and it has a minimum at a threshold value of 3e-3.

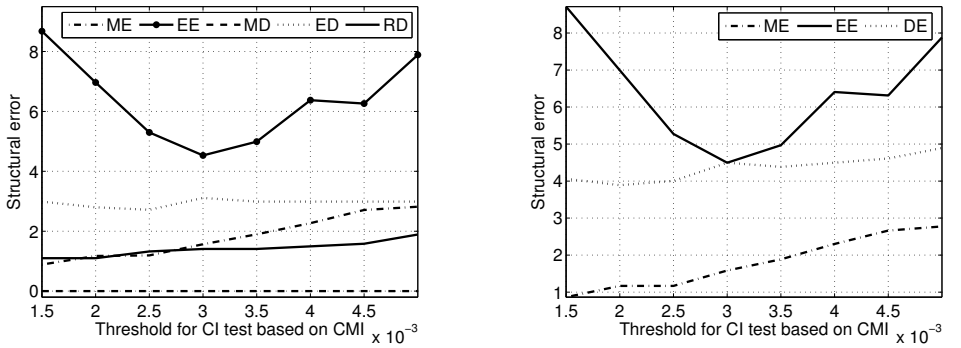


Figure 12: Structural errors of the RAI algorithm learning the Alarm network for different CMI thresholds as averaged over ten validation sets of 10,000 samples each. (a) Five types (ME, EE, MD, ED and RD) of structural errors, (b) EE, ME and DE errors, and (c) SHD error (mean and std).

In Figure 12b, we cast the three directional errors using the total directional error (DE), DE = ED + MD + RD, and plot this error together with the ME and EE errors. The impact of each error for increasing thresholds is now clearer; the contribution of the DE error is almost constant, that of the ME error increases with the threshold but is less than DE, and that of the EE error dominants for every threshold.

Tsamardinos et al. (2006a) suggested assessing the quality of a learned structure using the structural Hamming distance (SHD) metric, which is the sum of the five above errors. We plot in Figure 12c this error for the experiment with the Alarm network. Comparison of the threshold responsible for the minimum of the SHD error (2.5e-3) to that selected according to BDeu (4e-3 in Figure 11) shows only a small difference, especially as the maximum values of BDeu are obtained between thresholds of 2.5e-3 and 4e-3. This result motivates using the BDeu score, as measured on a validation set, as a criterion for finding good thresholds for RAI CI testing. Thresholds that are smaller than this range lead to too many pairs of variables that are wrongly identified as dependent and thus the edges between them are not removed, contributing to high EE errors (see, for example, Figure 12b). In addition, for thresholds higher than 3e-3, more edges are wrongly removed, contributing to high ME errors.

4.3. Learning the Alarm Network

For evaluating the correctness of learned BN structures, we used the Alarm network, which is widely accepted as a benchmark for structure learning algorithms, since the true graph for this problem is known. The RAI algorithm was compared to the PC, TPDA, GES, SC and MMHC algorithms using ten databases containing 10,000 random instances each sampled from the network.

Structural correctness can be measured using different scores. However, some of the scores suggested in the literature are not always accurate or related to the true structure. For example, Tsamardinos et al. (2006a), who examined the BDeu score (Heckerman et al., 1995) and KL divergence (Kullback and Leibler, 1951) in evaluating learned networks, noted that it is not known in practice to what degree the assumptions (e.g., a Dirichlet distribution of the hyperparameters) in the basis of the BDeu score hold. Moreover, usually such a score is used in both learning and evaluation of a structure; hence the score favors algorithms that use it in learning. Tsamardinos et al. (2006a) also mentioned that both scores do not rely on the true structure. Thus, they suggested the SHD metric, which is directly related to structural correctness, since it is the sum of the five errors of Section 4.2. Nevertheless, since SHD can be measured only when the true graph is known, scores such as BDeu and KL divergence are of great value in practical situations, for example, in classification problems like those examined in Section 4.5 in which the true graph is not known. These scores are also beneficial in the determination of algorithm parameters. For example, in Section 4.2 we measured BDeu scores of structures learned using different thresholds in order to select a good threshold for RAI CI testing.

Although SHD sums all five structural errors, we were first interested in examining the contribution of each individual error to the total error. Table 1 summarizes the five structural errors for each algorithm as averaged over 10 databases of 10,000 instances each sampled from the Alarm network. These databases are different from those validation databases used for threshold setting. The table also shows the total directional error, DE, which is the sum of the three directional errors. Table 1 demonstrates that the lowest EE and DE errors are achieved by the RAI algorithm and the lowest ME error is accomplished by the MMHC algorithm. Computing SHD shows the advantage of the RAI (3.5) algorithm over the PC (4.3), TPDA (9.5), GES (5.4), MMHC (13.1) and the SC (24.3) algorithms. Further, we propose such a table as Table 1 as a useful tool for the identification of the sources of structural errors of a given structure learning algorithm.

Note that the SHD error weighs each of the five error types equally. We believe that a score that weighs the five types based on their relative significance to structure learning will be a more accurate method to evaluate structural correctness; however, deriving such a score is a topic for future research.

Complexity was evaluated for each of the CB algorithms by measuring the number of CI tests employed for each order (condition set size) and the total number of log operations. The latter criterion is proportional to the total number of multiplications, divisions and logarithm evaluations that is required for calculating the CMI (Equation 2) during CI testing. Figure 13 depicts the average percentage (and number) of CI tests reduced by using the RAI algorithm compared to using the PC or TPDA algorithms for increasing sizes of the condition sets. The RAI algorithm reduces the number of CI tests of orders 1 and above required by the PC algorithm and those of orders 2 and above required by the TPDA algorithm. Moreover, the RAI algorithm completely avoids the use of CI tests of orders 4 and above and almost completely avoids CI tests of order 3

Table 1: Structural errors of several algorithms as averaged over 10 databases each containing 10,000 randomly generated instances of the Alarm network. The total directional error is the sum of three different directional errors, DE=ED+MD+RD, and the SHD error is DE+EE+ME. Bold font emphasizes the smallest error over all algorithms for each type of structural error.

	Extra Direction (ED)	Missing Direction (MD)	Reversed Direction (RD)	Directional Error (DE)	Extra Edge (EE)	Missing Edge (ME)	SHD
SC	1	9.5	4.6	15.1	4.7	4.5	24.3
MMHC	0.8	3.3	5.7	9.8	2.6	**0.7**	13.1
GES	0.1	0.6	1.2	1.9	2.7	0.8	5.4
TPDA	**0**	4.2	**0**	4.2	2.4	2.9	9.5
PC	**0**	**0**	0.8	0.8	2.5	1.0	4.3
RAI	**0**	**0**	0.3	**0.3**	**1.8**	1.4	**3.5**

compared to both the PC and TPDA algorithms. However, the RAI algorithm performs more CI tests of order 1 than the TPDA algorithm.

Figure 14 summarizes the total numbers of CI tests and log operations over different condition set sizes required by each algorithm. The RAI algorithm requires 46% less CI tests than the PC algorithm and 14% more CI tests (of order 1) than the TPDA algorithm. However, the RAI algorithm significantly reduces the number of log operations required by the other two algorithms. The PC or TPDA algorithms require, respectively, an additional 612% or 367% of the number of log operations required by the RAI algorithm. The reason for this substantial advantage of the RAI algorithm over both the PC and TPDA algorithms is the saving in CI tests of high orders (see Figure 13). These tests make use of large condition sets and thus are very expensive computationally.

4.4. Learning Known Networks

In addition to the state-of-art algorithms that were compared in Section 4.3, we include in this section the OR and GS algorithms. We compare the performance of the RAI algorithm to these algorithms by learning the structures of known networks employed in real decision support systems from a wide range of applications. We use known networks described in Tsamardinos et al. (2006a), which include the Alarm (Beinlich et al., 1989), Barley (Kristensen and Rasmussen, 2002), Child (Cowell et al., 1999), Hailfinder (Jensen and Jensen, 1996), Insurance (Binder et al., 1997), Mildew (Jensen and Jensen, 1996) and Munin (Andreassen et al., 1989) networks. All these networks may be downloaded from the Causal Explorer webpage. The Pigs, Link and Gene networks, which were also evaluated in Tsamardinos et al. (2006a), are omitted from our experiment due to memory and run-time limitations of the platform used in our experiment. These limitations are in the computation of the BDeu scoring function (part of the BNT toolbox) that is used for selecting a threshold for the RAI CI tests (Section 4.2).

The Casual Explorer webpage also contains larger networks that were created by tiling networks, such as the Alarm, Hailfinder, Child and Insurance, 3, 5 and 10 times.

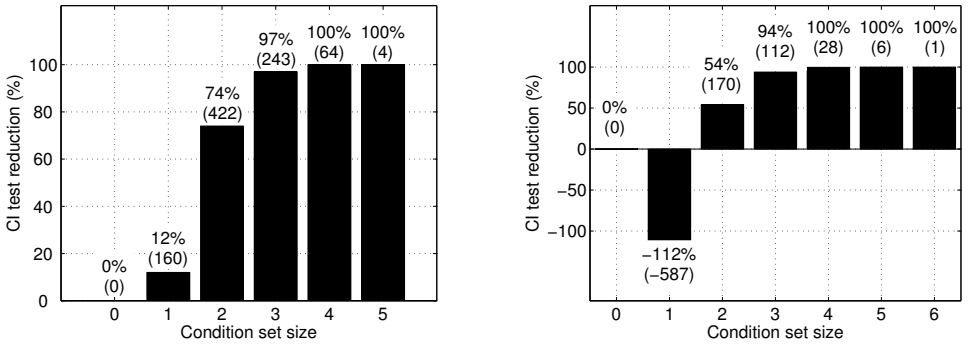


Figure 13: Average percentage (number) of CI tests reduced by using RAI compared to using (a) PC and (b) TPDA, as a function of the condition set size when learning the Alarm network.

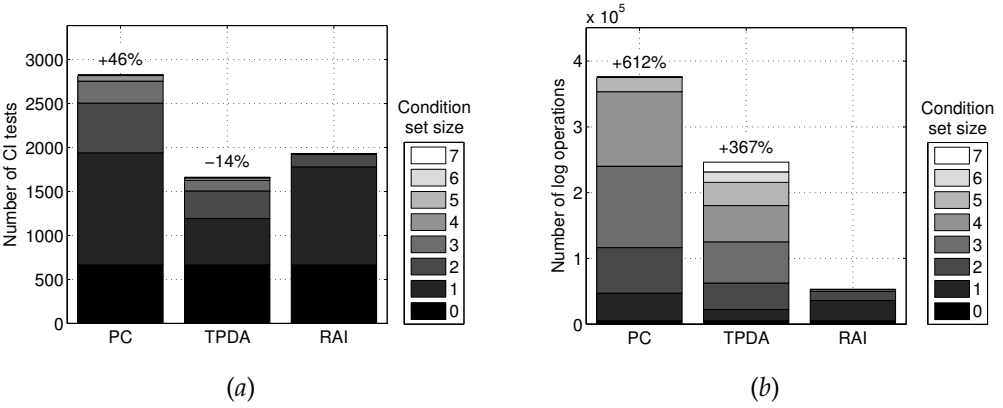


Figure 14: Cumulative numbers of (a) CI tests and (b) log operations required by PC, TPDA, and RAI for learning the Alarm network. Different gray shades represent different sizes of condition sets. Percentages on tops of the bars are with reference to the RAI algorithm.

Table 2: Nineteen networks with known structures that are used for the evaluation of the structure learning algorithms. The number that is attached to the network name (3, 5 or 10) indicates the number of tiles of this network. The # symbol on the first column represents the network ID for further use in the subsequent tables.

#	Network	# nodes	# edges	Max fan-in	Max fan-out
1	Alarm	37	46	4	5
2	Alarm 3	111	149	4	5
3	Alarm 5	185	265	4	6
4	Alarm 10	370	570	4	7
5	Barley	48	84	4	5
6	Child	20	25	2	7
7	Child 3	60	79	3	7
8	Child 5	100	126	2	7
9	Child 10	200	257	2	7
10	Hailfinder	56	66	4	16
11	Hailfinder 3	168	283	5	18
12	Hailfinder 5	280	458	5	18
13	Hailfinder 10	560	1017	5	20
14	Insurance	27	52	3	7
15	Insurance 3	81	163	4	7
16	Insurance 5	135	281	5	8
17	Insurance 10	270	556	5	8
18	Mildew	35	46	3	3
19	Munin	189	282	3	15

In the tiling method developed by Tsamardinos et al. (2006b), several copies (here 3, 5 and 10) of the same BN are tiled until reaching a network having a desired number of variables (e.g., Alarm5 has $5 \times 37 = 185$ variables). The method maintains the structural and probabilistic properties of the original network but allows the evaluation of the learning algorithm as the number of variables increases without increasing the complexity of the network. Overall, we downloaded and used nineteen networks, the most important details of which are shown in Table 2. Further motivation for using these networks and tiling is given in Tsamardinos et al. (2006a).

Throughout this experiment, we used for each network the same training and test sets as used in Tsamardinos et al. (2006a), so we could compare the performance of the RAI to all the algorithms reported in Tsamardinos et al. (2006a). The data in the Causal Explorer webpage are given for each network using five training sets and five test sets with 500, 1000 and 5,000 samples each. We picked and downloaded the data sets with the smallest sample size (500), which we believe challenge the algorithms the most. All the reported results for a network and a learning algorithm in this sub-section are averages over five experiments in which a different training set was used for training the learning algorithm and a different test set was used for testing this algorithm.

The RAI algorithm was run by us. CMI thresholds for CI testing corresponded to the maximum BDeu values were obtained in five runs using five validation sets independent of the training and test sets, and performances were averaged over the five validation sets. We note that the thresholds selected according to the maximum BDeu values (Section 4.2) also led to the lowest SHD errors. The OR algorithm was examined with a maximum number of parents allowed for a node (k) of 5, 10 and 20 and allowed run-time that is one and two times the time used by MMHC on the corresponding data set (OR1 and OR2, respectively). The SC algorithm was evaluated with $k = 5$ and $k = 10$ as recommended by its authors. Motivation for using these parameter values and parameter values used by the remaining algorithms are given in Tsamardinos et al. (2006a).

Following Tsamardinos et al. (2006a), we normalized all SHD results with the SHD results of the MMHC algorithm. For each network and algorithm, we report on the average ratio over the five runs. The normalized SHDs are presented in Table 3. A ratio smaller (larger) than 1 indicates that the algorithm learns a more (less) accurate structure than that learned using the MMHC algorithm. In addition, we average the ratios over all nineteen databases similarly to Tsamardinos et al. (2006a). Based on these averaged ratios, Tsamardinos et al. (2006a) found the MMHC algorithm to be superior to the PC, TPDA, GES, OR and SC algorithms with respect to SHD. Table 3 shows that the RAI algorithm is the only algorithm that achieves an average ratio that is smaller than 1, which means it learns structures that on average are more accurate than those learned by MMHC, and thus also more accurate than those learned by all other algorithms. Note the difference in SHD values for Alarm between Table 3 (as measured in Tsamardinos et al., 2006a, on databases of 500 samples) and Table 1 (as measured by us on databases of 10,000 samples).

Next, we compared the run-times of the algorithms in learning the nineteen networks. We note that the run-time of a structure learning algorithm depends, besides on its implementation, on the number of statistical calls (Tsamardinos et al., 2006a) it performs (e.g., CI tests in CB algorithms). For CB algorithms it also depends on the orders of the CI tests and the number of states of each variable that is included in the condition set. The run-time for each algorithm learning each network is presented in Table 4. Following Tsamardinos et al. (2006a), we normalized all run-time results with

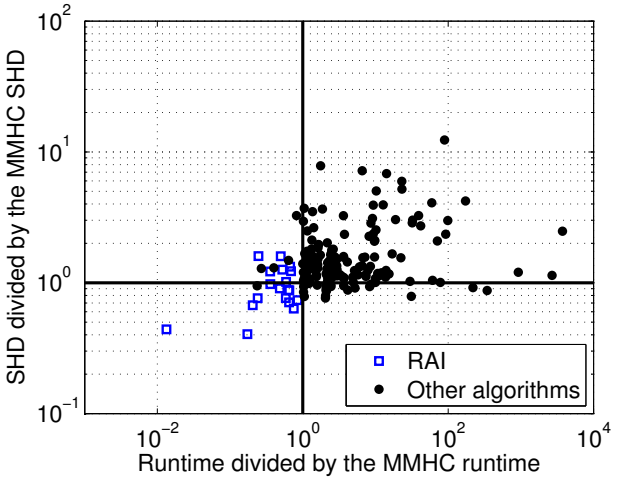
Table 3: Algorithm SHD errors normalized with respect to the MMHC SHD error for the nineteen networks detailed in Table 2. Average (avg.) for an algorithm is over all networks. Blank cells represent jobs that Tsamardinos et al. (2006a) reported that refused to run or did not complete their computations within two days running time.

#	MMHC	OR1 $k=5$	OR1 $k=10$	OR1 $k=20$	OR2 $k=5$	OR2 $k=10$	OR2 $k=20$	SC $k=5$	SC $k=10$	GS	PC	TPDA	GES	RAI
1	1.00	1.23	1.39	1.67	1.05	1.02	1.40	1.63	1.66	2.02	3.66	2.34		1.23
2	1.00	1.85	1.95	1.96	1.78	1.77	1.80	1.57	1.57	2.26	2.49	3.94		1.26
3	1.00	1.59	1.61	1.63	1.48	1.63	1.69	1.32	1.35	2.10	2.35	3.10		1.02
4	1.00	1.46	1.52	1.53	1.49	1.52	1.57	1.18		2.09		2.72		0.87
5	1.00	1.03	1.05	1.08	0.98	0.97	0.99	1.15		1.16	12.34	1.44	0.92	0.67
6	1.00	1.38	1.30	1.15	1.25	1.24	1.15	1.48	1.56	0.79	3.26	7.18	0.79	1.60
7	1.00	0.99	1.06	1.03	0.87	0.86	1.01	0.95	0.97	0.94	2.95	5.03	1.20	1.22
8	1.00	1.45	1.74	1.69	0.89	1.10	0.99	0.88	0.93	1.15	3.71	6.82	2.48	1.59
9	1.00	2.12	1.40	1.81	1.42	1.44	1.45	1.08	1.12	1.19	3.49	5.96		1.33
10	1.00	1.01	0.99	1.03	0.99	0.99	1.01	0.96		0.99	2.64	2.36	1.14	0.41
11	1.00	1.33	1.34	1.34	1.27	1.26	1.28	1.10		1.01	3.92	3.01		0.71
12	1.00	1.40	1.41	1.42	1.30	1.30	1.28	1.12		1.01	5.20	3.26		0.76
13	1.00	1.33	1.33	1.34	1.34	1.29	1.33	1.10		1.02		2.99		0.74
14	1.00	1.04	0.93	0.85	0.95	0.79	0.76	1.33	1.17	1.20	3.26	2.54	1.01	0.76
15	1.00	1.08	1.06	1.25	1.04	1.14	1.15	1.26	1.33	1.57	4.09	3.04		0.98
16	1.00	1.25	1.24	1.12	1.13	1.15	1.17	1.24	1.25	1.59	4.22	2.86		0.91
17	1.00	1.30	1.29	1.31	1.19	1.13	1.24	1.18	1.24	1.55		2.87		0.88
18	1.00	1.09	1.11	1.10	1.10	1.12	1.07	1.04		0.91	7.83	2.08	0.87	0.63
19	1.00	1.09	1.16	1.06	1.17			0.95		1.30		1.29		0.44
avg.	1.00	1.32	1.31	1.33	1.19	1.21	1.24	1.19	1.29	1.36	4.36	3.41	1.20	0.95

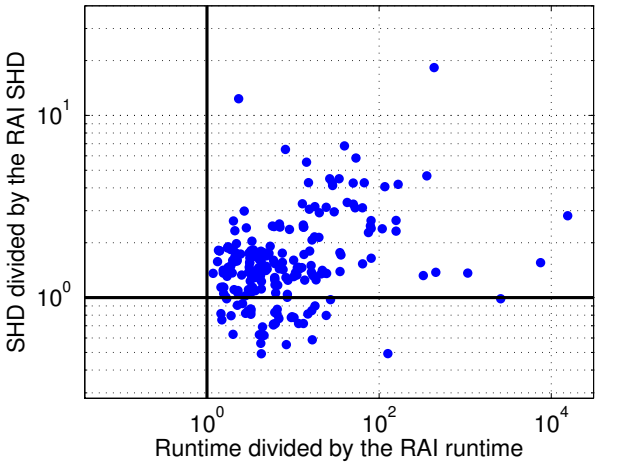
the run-time results of the MMHC algorithm and report on the average ratio for each algorithm and network over five runs. The run-time ratios for all algorithms except that for the RAI were taken from the Causal Explorer webpage. The ratio for the RAI was computed after running both the RAI and MMHC algorithms on our platform using the same data sets. According to Tsamardinos et al. (2006a), MMHC is the fastest algorithm among all algorithms (except RAI). Table 4 shows that RAI was the only algorithm that achieved an average ratio smaller than 1, which means it is the new fastest algorithm. The RAI average run-time was between 2.1 (for MMHC) and 2387 (for GES) times shorter than those of all other algorithms. Perhaps part of the inferiority of GES with respect to run-time can be related (Tsamardinos et al., 2006a) to many optimizations suggested in Chickering (2002) that were not implemented in Tetrad 4.3.1 that was used by Tsamardinos et al. (2006a) affecting their, and thus also our, results.

Accounting for both error and time, we plot in Figure 15 the SHD and run-time for all nineteen networks normalized with respect to either the MMHC algorithm (Figure 15a) or the RAI algorithm (Figure 15b). Figure 15 demonstrates that the advantage of RAI over all other algorithms is evident for both the SHD error and the run-time.

It is common to consider the statistical calls performed by an algorithm of structure learning as the major criterion of computational complexity (efficiency) and a major contributor to the algorithm run-rime. In CB algorithms (e.g., PC, TPDA and RAI), the statistical calls are due to CI tests, and in S&S algorithms (e.g., GS, GES, SC, OR) the calls are due to the computation of the score. Hybrid algorithms (e.g., MMHC) have both types of calls. In Table 5, we compare the numbers of calls for statistical



(*a*)



(*b*)

Figure 15: Normalized SHD vs. normalized run-time for all algorithms learning all networks. (a) Normalization is with respect to the MMHC algorithm (thus MMHC results are at (1,1)) and (b) normalization is with respect to the RAI algorithm (thus RAI results are at (1,1)). The points in the graph correspond to 19 networks (average performance over 5 runs) and $14 - 1 = 13$ algorithms.

Table 4: Algorithm run-times normalized with respect to the MMHC run-time for the nineteen networks detailed in Table 2. Average (avg.) for an algorithm is over all networks. Blank cells represent jobs that Tsamardinos et al. (2006a) reported that refused to run or did not complete their computations within two days running time.

#	MMHC	OR1 $k=5$	OR1 $k=10$	OR1 $k=20$	OR2 $k=5$	OR2 $k=10$	OR2 $k=20$	SC $k=5$	SC $k=10$	GS	PC	TPDA	GES	RAI
1	1.00	1.14	1.00	1.07	2.24	2.22	2.33	1.75	16.93	2.17	1.87	3.74		0.69
2	1.00	1.62	1.65	1.64	2.51	2.53	2.63	7.15	9.71	8.16	1.15	12.75		0.52
3	1.00	1.21	1.32	1.33	2.35	2.41	2.48	6.01	6.54	9.80	92.64	9.11		0.59
4	1.00	1.38	1.61	1.43	2.87	2.93	2.77	13.85		71.15		41.81		0.65
5	1.00	1.26	1.24	1.21	2.29	2.42	2.36	7.36		2.74	89.28	4.10	219.5	0.20
6	1.00	1.61	1.61	1.53	2.39	2.34	3.25	0.64	6.71	1.05	0.82	6.56	31.12	0.25
7	1.00	1.15	1.14	1.06	2.12	2.10	2.18	3.66	8.64	2.44	1.02	10.27	921	0.36
8	1.00	1.12	1.14	1.13	2.10	2.19	2.29	4.16	8.31	5.76	1.05	14.19	3738	0.50
9	1.00	1.34	1.05	1.32	2.20	2.28	2.45	9.97	11.08	12.10	1.36	22.99		0.67
10	1.00	1.20	1.22	1.21	2.31	2.29	2.28	1.58		1.04	1.42	9.31	2690	0.17
11	1.00	1.13	1.15	1.14	2.15	2.21	2.27	4.88		4.96	9.32	32.39		0.65
12	1.00	1.11	1.15	1.17	2.24	2.27	2.19	7.39		10.01	23.14	39.22		0.58
13	1.00	1.18	1.19	1.15	2.94	2.61	2.74	13.77		29.84		99.00		0.85
14	1.00	1.02	1.03	1.03	2.09	2.06	2.05	1.26	15.36	1.02	3.62	10.19	78.06	0.24
15	1.00	1.09	1.13	1.18	2.25	2.38	2.21	2.96	8.50	3.63	59.50	18.87		0.36
16	1.00	1.49	1.48	1.54	2.97	2.95	2.96	5.15	7.88	3.63	173.3	8.67		0.48
17	1.00	1.19	1.12	1.20	2.30	2.35	2.40	10.73	13.95	22.34		32.00		0.64
18	1.00	2.46	2.43	2.55	3.68	3.46	3.68	61.04		5.23	1.76	9.67	343.7	0.75
19	1.00	1.05	1.07	1.08	2.09			0.24		0.40		0.27		0.01
avg.	1.00	1.30	1.30	1.31	2.43	2.45	2.53	8.61	10.33	10.39	30.75	20.27	1146	0.48

tests performed by the RAI algorithm and computed by us to those of the MMHC, GS, PC and TPDA, as computed in Tsamardinos et al. (2006a), and downloaded from the Causal Explorer webpage. We find that for all networks the RAI algorithm performs fewer calls for statistical tests than all other algorithms. On average over all networks, the RAI algorithm performs only 53% of the calls for statistical tests performed by the MMHC algorithm, which is the algorithm that required the fewest calls of all algorithms examined in Tsamardinos et al. (2006a). Figure 16 demonstrates this advantage of RAI over MMHC graphically using a scatter plot. All points below the $x = y$ line represent data sets for which the numbers of calls for statistical tests of MMHC are larger than those of RAI.

Evaluating the statistical significance of the results in Tables 3–5 using Wilcoxon signed-ranks test (Demšar, 2006) with a confidence level of 0.05, we find the SHD errors of RAI and MMHC to be not significantly different from each other; however, the RAI run-times and numbers of statistical calls are significantly shorter than those of the MMHC algorithm.

In continuation to Section 4.1, we further analyzed the complexity of RAI (as measured by the numbers of CI tests performed) according to the CI test orders and the graph size. However, here we used real rather than synthetic data. We examined the numbers of tests as performed for different orders for the Child, Insurance, Alarm and Hailfinder networks and their tiled networks. Using the tiled networks (Tsamardinos et al., 2006b), we could examine the impact of graph size on the number of tests. Figure 17 shows the cumulative percentage of CI tests for a specific order out of the

Table 5: Number of statistical calls performed by each algorithm normalized by the number of statistical calls performed by the MMHC algorithm for the nineteen networks detailed in Table 2. Average (avg.) for an algorithm is over all networks. Blank cells represent jobs that Tsamardinos et al. (2006a) reported that refused to run or did not complete their computations within two days running time.

#	MMHC	GS	PC	TPDA	RAI
1	1.00	2.42	9.95	1.94	0.81
2	1.00	3.78	2.51	3.34	0.57
3	1.00	4.44	1499.22	3.02	0.67
4	1.00	5.12		2.64	0.75
5	1.00	1.96	2995.87	1.58	0.34
6	1.00	1.32	3.61	2.92	0.21
7	1.00	2.49	4.61	2.97	0.39
8	1.00	3.25	4.40	3.17	0.51
9	1.00	3.91	5.43	3.13	0.64
10	1.00	1.75	36.54	1.93	0.30
11	1.00	2.57	340.44	1.83	0.72
12	1.00	3.07	1033.86	1.87	0.67
13	1.00	3.40		1.85	0.77
14	1.00	1.32	40.57	2.97	0.27
15	1.00	2.35	1082.45	2.71	0.39
16	1.00	3.12	5143.51	2.97	0.49
17	1.00	4.25		3.20	0.63
18	1.00	3.38	10.78	3.49	0.59
19	1.00	1.75		0.91	0.30
avg.	1.00	2.93	814.25	2.55	0.53

total number of CI tests performed for each network. The figure demonstrates that the percentages of CI tests performed decrease with the CI test order and become small for orders higher than the max fan-in of the network (see Table 2). These percentages also decrease with the numbers of nodes in the network (validated on the tiled networks). This is due to a faster increase of the number of low-order CI tests compared with the number of high-order CI tests as the graph size increases for all networks except for Hailfinder. For Hailefinder (Figure 17d), the threshold for the network was different from those of the tiled networks. This led to an increase in the percentage of high-order CI tests and a decrease in CI tests of order 0 when comparing the Hailfinder network to its tiled versions. For all the tiled Alarm networks (Figure 17c), CI tests of order 0 nearly sufficed for learning the network. Overall, the results support our preliminary results with synthetic data and "perfect" CI tests (Section 4.1). Thus, we can conclude that as the graph size increases, the RAI algorithm requires relatively fewer CI tests of high orders, especially of orders higher than the max fan-in, than tests of low orders. This result enhances the attractiveness in applying the RAI algorithm also to large problems.

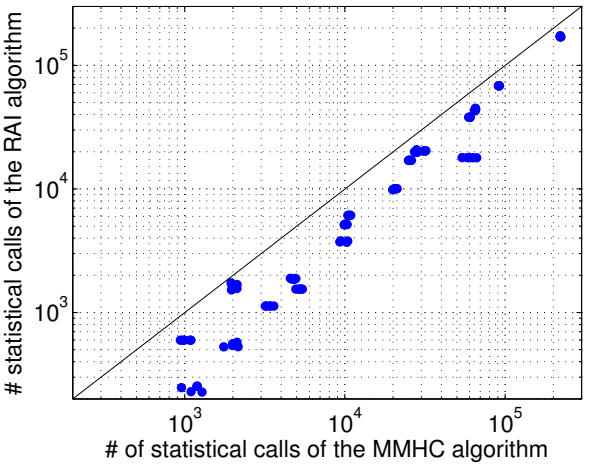


Figure 16: Number of statistical calls performed by the RAI algorithm vs. the number of statistical calls performed by the MMHC algorithm for all networks and data sets examined in this sub-section (5 data sets $\times$ 19 networks = 95 points).

4.5. Structure Learning for General BN Classifiers

Classification is one of the most fundamental tasks in machine learning (ML), and a classifier is primarily expected to achieve high classification accuracy. The Bayesian network classifier (BNC) is usually not considered as an accurate classifier compared to state-of-the-art ML classifiers, such as the neural network (NN) and support vector machine (SVM). However, the BNC has important advantages over the NN and SVM models. The BNC enhances model interpretability by exhibiting dependences, independences and causal relations between variables. It also allows the incorporation of prior knowledge during model learning so as to select a better model or to improve the estimation of its data-driven parameters. Moreover, the BNC naturally performs feature selection as part of model construction and permits the inclusion of hidden nodes that increase model representability and predictability. In addition, the BN has a natural way of dealing with missing inputs by marginalizing hidden variables. Finally, compared to NN and SVM, BNC can model very large, multi-class problems with different types of variables. These advantages are important in real-world classification problems, since they provide many insights into the problem at hand that are beyond the pure classification decisions provided by NN and SVM.

We evaluated the RAI complexity, run-time and accuracy when applied to learning a general BN classifier (Cheng and Greiner, 1999; Friedman et al., 1997) in comparison to other algorithms of structure learning using nineteen databases of the UCI Repository (Newman et al., 1998) and Kohavi and John (1997). These databases are detailed in Table 6 with respect to the numbers of variables, classes and instances in each database. All databases were analyzed using a CV5 experiment, except large databases (e.g., "chess", "nursery" and "shuttle"), which were analyzed using the holdout methodology and the common division to training and test sets (Newman et al., 1998; Friedman et al., 1997; Cheng et al., 1997) as detailed in Table 6. Continuous variables were discretized

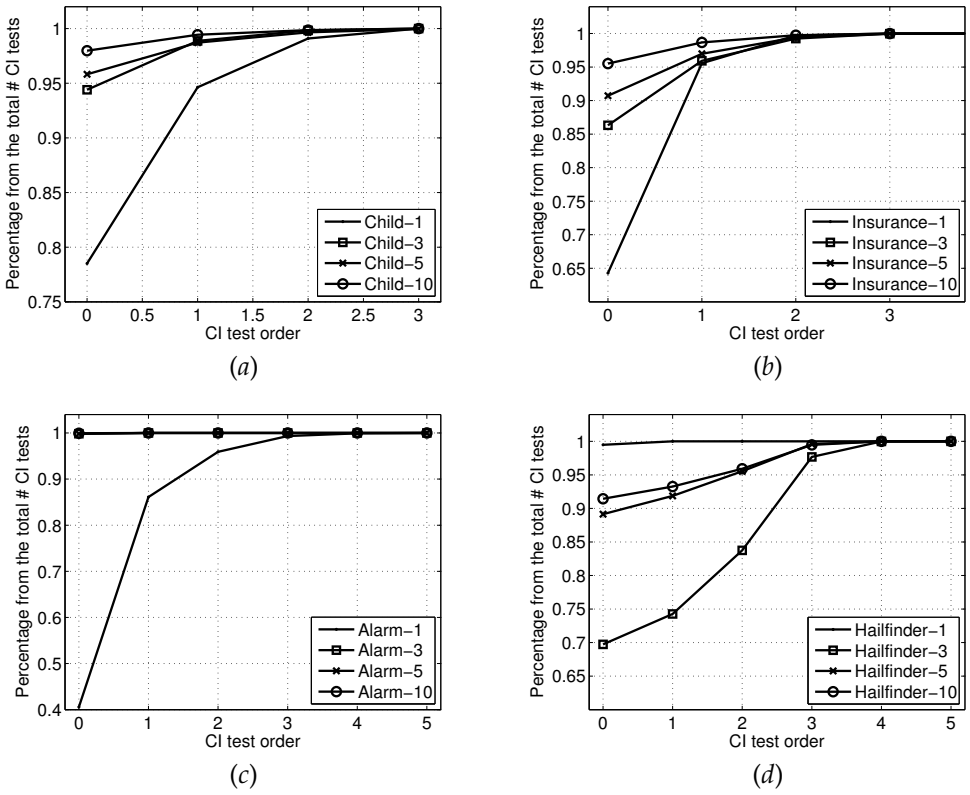


Figure 17: Cumulative percentages of CI tests out of the total numbers of tests for increasing orders as performed by the RAI algorithm for the (a) Child, (b) Insurance, (c) Alarm, and (d) Hailfinder networks including their tiled networks.

using the MLC++ library (Kohavi et al., 1994) and instances with missing values were removed, as is commonly done.

Table 6: Databases of the UCI repository (Newman et al., 1998) and of Kohavi and John (1997) used for evaluating the accuracy of a classifier learned using the RAI algorithm.

Database	# variables	# classes	# instances	Test methodology	# training instances	# test instances
australian	14	2	690	CV5	552	138
breast	9	2	683	CV5	544	136
car	6	4	1728	CV5	1380	345
chess	36	2	3196	holdout	2130	1066
cleve	11	2	296	CV5	236	59
cmc	9	3	1473	CV5	1176	294
corral	6	2	128	CV5	100	25
crx	15	2	653	CV5	520	130
flare C	10	9	1389	CV5	1108	277
iris	4	3	150	CV5	120	30
led7	7	10	3200	CV5	2560	640
mofn 3-7-10	10	2	1324	holdout	300	1024
nursery	8	5	12960	holdout	8640	4320
shuttle (s)	8	7	5800	holdout	3866	1934
tic-tac-toe	9	2	958	CV5	764	191
vehicle	18	4	846	CV5	676	169
vote	16	3	435	CV5	348	87
wine	13	3	178	CV5	140	35
zoo	16	7	101	CV5	80	20

Generally for this sub-section, CI tests for RAI and PC were carried out using the χ^2 test (Spirtes et al., 2000) and those for TPDA using the CMI independence test (Equation 2). However, CI tests for RAI and PC for the "corral", "nursery" and "vehicle" databases were carried out using the CMI independence test. In the case of the large "nursery" database, the need to use the CMI test was due to a Matlab memory limitation in the completion of the χ^2 test using the BNT structure learning package (Leray and François, 2004). In the case of the "corral" and "vehicle" databases, the smallness of the database, together with either the large numbers of classes, variables or states for each variable, led to low frequencies of instances for many combinations of variable states. In this case, the implementation of the χ^2 test assumes variable dependence (Spirtes et al., 2000) that prevents the CB (PC, TPDA and RAI) algorithms from removing edges regardless of the order of the CI test, leading to erroneous decisions. Another test of independence, which is reported to be more reliable and robust, especially for small databases or large numbers of variables (Dash and Druzdzel, 2003), may constitute another solution in these cases.

Thresholds for the CI tests of the CB algorithms and parameter values for all other algorithms were chosen for each algorithm and database so as to maximize the classification accuracy on a validation set selected from the training set or based on the

recommendation of the algorithm authors or of Tsamardinos et al. (2006a). Although using a validation set decreases the size of the training set, it also eliminates the chance of selecting a threshold or a parameter that causes the model to overfit the training set at the expense of the test set. If several thresholds/parameters were found suitable for an algorithm, the threshold/parameter chosen was that leading to the fewest CI tests (in the case of CB algorithms). For GES and GS there are no parameters to set (except the equivalent sample size for the BDeu), and for MMHC we used the selections used by the authors in all their experiments.

Finally, parameter learning was performed by maximum likelihood estimation. Since we were interested in structure learning, no attempt was made to study estimation methods other than this simple and most popular generative method (Cooper and Herskovits, 1992; Heckerman, 1995; Yang and Chang, 2002). Nevertheless, we note that discriminative models for parameter learning have recently been suggested (Pernkopf and Bilmes, 2005; Roos et al., 2005). These models show an improvement over generative models when estimating the classification accuracy (Pernkopf and Bilmes, 2005). We expect that any improvement in classification accuracy gained by using parameter learning other than maximum likelihood estimation will be shared by classifiers induced using any algorithm of structure learning; however, the exact degree of improvement in each case should be further evaluated.

Complexity of the RAI algorithm was measured by the number of CI tests employed for each size of the condition set and the cumulative run-time of the CI tests. These two criteria of complexity were also measured for the PC algorithm, since both the RAI and PC algorithms use the same implementation of CI testing. Table 7 shows the average number and percentage of CI tests reduced by the RAI algorithm compared to the PC algorithm for different CI test orders and each database. An empty entry in the table means that no CI tests of this order are required. A 100% cut in CI tests for a specific order means that RAI does not need any of the CI tests employed by the PC algorithm for this order (e.g., orders 2 and above for the "led7" database). It can be seen that for almost all databases examined, the RAI algorithm avoids most of the CI tests of orders two and above that are required by the PC algorithm (e.g., the "chess" database). Table 7 also shows the reduction in the CI test run-time due to the RAI algorithm in comparison to the PC algorithm for all nineteen databases examined; except for the "australian" database, the cut is measured in tens of percentages for all databases and for six databases this cut is higher than 70%. Run-time differences between algorithms may be the result of different implementations. However, since in our case the run-time is almost entirely based on the number and order of CI tests and RAI has reduced most of the PC CI tests, especially those of high orders that are expensive in run-time, we consider the above run-time reduction results to be significant.

Classification accuracy using a BNC has recently been explored extensively in the literature (Friedman et al., 1997; Grossman and Domingos, 2004; Kontkanen et al., 1999; Pernkopf and Bilmes, 2005; Roos et al., 2005). By restricting the general inference task of BN to inference performed on the class variable, we turn a BN into a BNC. First, we use the training data to learn the structure and then transform the pattern outputted by the algorithm into a DAG (Dor and Tarsi, 1992). Thereafter, we identify the class node Markov blanket and remove from the graph all the nodes that are not part of this blanket. Now, we could estimate the probabilities comprising the class node posterior probability, $P(C|\boldsymbol{X})$, where $\boldsymbol{X}$ is the set of the Markov blanket variables. During the test, we inferred the state c of the class node C for each test instantiation, $\boldsymbol{X} = \boldsymbol{x}$, using the estimated posterior probability. The class $\hat{c}$ selected was the one that maximized the

Table 7: Average number (and percentage) of CI tests reduced by the RAI algorithm compared to the PC algorithm for different databases and CI test orders and the cut (%) in the total CI test run-time.

Database	CI test order									Run-time cut (%)
	0	1		2		3		4		
australian	0 (0)	3.8	(34.4)							6.05
breast	0 (0)	107.2	(54.8)	35	(99.1)					71.87
car	0 (0)	16	(100)	11.2	(100)	3.2	(100)			91.10
chess	0 (0)	2263	(76.3)	2516	(89)	581	(94)	249	(100)	80.65
cleve	0 (0)	12.4	(63)							39.60
cmc	0 (0)	10.2	(10.9)	8	(32.5)					14.22
corral	0 (0)	22.4	(100)	26	(100)	3.6	(100)			87.94
crx	0 (0)	8.8	(49.6)							25.25
flare C	0 (0)	16	(39.6)	3	(100)					20.38
iris	0 (0)	2	(40)							19.10
led7	0 (0)	46.2	(45.7)	105	(100)	140	(100)	105	(100)	91.74
mofn 3-7-10	0 (0)	17	(100)	4	(100)					67.70
nursery	0 (0)	20	(100)	30	(100)	20	(100)	5	(100)	89.70
shuttle (s)	0 (0)	1.4	(0.7)	95.8	(43.8)	117.6	(49.3)	83.6	(56.0)	38.94
tic-tac-toe	0 (0)	53.2	(27.1)	56.6	(48.6)	1.8	(51.4)			36.52
vehicle	0 (0)	-12.4	(-2.9)	32.6	(20.4)	-5.8	(-14.0)	3.4	(27.4)	13.15
vote	0 (0)	24.2	(21.9)	17.2	(98.1)	6.4	(100)	1	(100)	46.06
wine	0 (0)	25.8	(41.0)	44.2	(67.6)	40.6	(82.4)	19	(96.7)	29.11
zoo	0 (0)	82	(27.8)	365.8	(29.6)	1033.4	(27.7)	1928.6	(25.6)	13.63

posterior probability, meaning that $\hat{c} = \arg\max_c P(C = c|\boldsymbol{X} = \boldsymbol{x})$. By comparing the class maximizing the posterior probability and the true class, we could compute the classification accuracy.

In Table 8 we compared the classification accuracy due to the RAI algorithm to those due to the PC, TPDA, GES, MMHC, SC and NBC algorithms. We note the overall advantage of the RAI algorithm, especially for large databases. Since the reliability of the CI tests increased with the sample size, it seems that RAI benefits from this increase more than the other algorithms and excels in classifying large databases. RAI, when compared to the other structure learning algorithms, yielded the best classifiers on six ("flare C", "nursery", "led7", "mofn", "tic-tac-toe" and "vehicle") of the ten largest databases and among the best classifiers on the remaining four ("shuttle", "chess", "car" and "cmc") large databases. The other CB algorithms — PC and TPDA — also showed here, and in Tsamardinos et al. (2006a), better results on the large databases. However, the CB algorithms are less accurate on very small databases (e.g., "wine" and "zoo").

Overall, RAI was the best algorithm on 7 databases compared to 5, 2, 5, 4, 5 and 5 databases for the PC, TPDA, GES, MMHC, SC and NBC algorithms, respectively. RAI was the worst classifier on only a single database, whereas the PC, TPDA, GES, MMHC, SC and NBC algorithms were the worst classifiers on 2, 4, 6, 2, 2 and 7 databases, respectively. We believe that the poor results of the GES and MMHC algorithms on the "nursery" database may be attributed to the fact that these algorithms find the class node C as a child of many other variables, making the estimation of $P(C|\boldsymbol{X})$ unreliable due to

Table 8: Mean (and standard deviation for CV5 experiments) of the classification accuracy of the RAI algorithm in comparison to those of the PC, TPDA, GES, MMHC, SC and NBC algorithms. **Bold** and *italic* fonts represent, respectively, the best and worst classifiers for a database.

Database	PC	TPDA	GES	MMHC	SC	NBC	RAI
australian	85.5 (0.5)	85.5 (0.5)	*83.5* (2.1)	**86.2** (1.5)	85.5 (1.2)	85.9 (3.4)	85.5 (0.5)
breast	95.5 (2.0)	*94.4* (2.7)	96.8 (1.1)	97.2 (1.2)	96.5 (0.8)	**97.5** (0.8)	96.5 (1.6)
car	84.3 (2.6)	84.5 (0.6)	*81.5* (2.3)	90.2 (2.0)	**93.8** (1.1)	84.7 (1.3)	92.9 (1.1)
chess	93.1	90.1	**97.0**	94.1	92.5	*87.1*	93.5
cleve	76.7 (7.2)	*72.0* (10.7)	79.4 (5.7)	82.1 (4.5)	**83.5** (5.7)	**83.5** (5.2)	81.4 (5.4)
cmc	50.9 (2.3)	46.4 (2.1)	*46.3* (1.5)	48.6 (2.6)	49.7 (2.5)	**51.3** (1.3)	51.1 (3.2)
corral	**100** (0)	88.2 (6.4)	**100** (0)	**100** (0)	**100** (0)	*85.2* (7.3)	**100** (0)
crx	86.4 (2.6)	**86.7** (3.4)	*82.2* (6.4)	**86.7** (1.7)	**86.7** (3.4)	86.2 (2.8)	86.4 (2.6)
flare C	**84.3** (2.5)	**84.3** (2.4)	**84.3** (2.5)	**84.3** (2.5)	**84.3** (2.5)	*77.7* (3.1)	**84.3** (2.5)
iris	**96.0** (4.3)	93.3 (2.4)	**96.0** (4.3)	94.0 (3.6)	*92.7* (1.5)	94.0 (4.3)	93.3 (2.4)
led7	73.3 (1.8)	*72.9* (1.5)	*72.9* (1.5)	*72.9* (1.5)	*72.9* (1.5)	*72.9* (1.5)	**73.6** (1.6)
mofn 3-7-10	81.4	90.8	*79.8*	90.5	91.9	89.8	**93.2**
nursery	**72.0**	64.7	33.3	*29.3*	30.3	66.0	**72.0**
shuttle (s)	98.4	*96.3*	**99.5**	99.2	99.2	98.8	99.2
tic-tac-toe	74.7 (1.4)	72.2 (3.8)	69.9 (2.8)	71.1 (4.2)	70.4 (4.7)	*69.6* (3.1)	**75.6** (1.9)
vehicle	63.9 (3.3)	65.6 (2.8)	64.1 (11.2)	69.3 (1.5)	64.8 (9.1)	*62.0* (4.0)	**70.2** (2.8)
vote	**95.9** (1.5)	95.4 (2.1)	94.7 (2.8)	95.6 (2.2)	93.1 (2.2)	*90.6* (3.3)	95.4 (1.6)
wine	*85.4* (7.8)	97.8 (3.0)	98.3 (2.5)	98.3 (2.5)	98.3 (2.5)	**98.9** (1.5)	87.1 (5.9)
zoo	*89.0* (8.8)	96.1 (2.2)	96.0 (2.3)	93.1 (4.5)	95.9 (6.9)	**96.3** (3.8)	*89.0* (8.79)
average	83.5	83.0	*81.9*	83.3	83.3	83.1	**85.3**
std	12.7	13.8	18.4	18.4	18.4	13.3	12.3

the curse-of-dimensionality. The structures learned by the other algorithms required a smaller number of such connections and thereby reduced the curse.

In addition, we averaged the classification accuracies of the algorithms over the nineteen databases. Averaging accuracies over databases has no meaning in itself except that the average accuracies over many different problems of different algorithms may infer about the relative expected success of the algorithms in other classification problems. It is interesting to note that although the different algorithms in our study showed different degrees of success on various databases, most of the algorithms (i.e., PC, TPDA, MMHC, SC and NBC) achieved almost the same average accuracy (83.0%-83.5%). The GES average accuracy was a little inferior (81.9%) to that of the above algorithms, and the average accuracy of the RAI (85.3%) was superior to that of all algorithms. Concerning the standard deviation of the classification accuracy, RAI outperformed all classifiers implying to the robustness of the RAI-based classifier.

Superiority of one algorithm over another algorithm for each database was evaluated with a statistical significance test (Dietterich, 1998). We used a single-sided t-test to evaluate whether the mean difference between any pair of algorithms as measured on the five folds of the CV5 test was greater than zero. Table 9 summarizes the statistical significance results, measured at a significance level of 0.05, for any two classifiers and each database examined using cross validation. The number in each cell of Table 9

Table 9: Statistical significance using a t-test for the classification accuracy results of Table 8. For a given database, each cell indicates the number of algorithms found to be inferior at a significance level of 0.05 to the algorithm above the cell.

Databse	PC	TPDA	GES	MMHC	SC	NBC	RAI
australian	1	1	0	1	0	1	1
breast	0	0	0	2	0	3	0
car	1	1	0	4	6	1	5
cleve	0	0	0	1	3	3	2
cmc	4	0	0	1	2	2	3
corral	2	0	2	2	2	0	2
crx	0	0	0	0	0	0	0
flare C	1	1	1	1	1	0	1
iris	1	0	1	0	0	0	0
led7	0	0	0	0	0	0	5
tic-tac-toe	3	2	0	0	0	0	5
vehicle	0	1	0	3	0	0	3
vote	2	2	1	3	0	0	1
wine	0	2	2	2	2	2	0
zoo	0	0	0	0	2	0	0
total	15	10	7	20	18	12	**28**
average	1.00	0.67	0.47	1.33	1.20	0.8	**1.87**

describes — for the corresponding algorithm and database — the number of algorithms that are inferior to that algorithm for that databases. A "0" value indicates that the algorithm is either inferior to all the other algorithms or not significantly superior to any of them. For example, for the "car" database the PC, TPDA, GES, MMHC, SC, NBC and RAI algorithms were significantly superior to 1, 1, 0, 4, 6, 1 and 5 other algorithms, respectively. In total, the superiority of the RAI algorithm over the other algorithms was statistically significant 28 times, with an average of 1.87 algorithms per database. The second and third best algorithms were the MMHC and SC algorithms, with a total of 20 and 18 times of statistically significant superiority and averages of 1.33 and 1.2 per database, respectively. The least successful classifier, according to Tables 8 and 9, was the one that is learned using GES. We believe that this inferiority arises from the assumptions on the type of probabilities and their parameters made by the GES algorithm when computing the BDeu score (Heckerman et al., 1995), assumptions that probably do not hold for the examined databases.

Although this methodology of statistical tests between pairs of classifiers is the most popular in the machine learning community, there are other methodologies that evaluate statistical significance between several classifiers on several databases simultaneously. For example, Demšar (2006), recently suggested using Friedman test (Friedman, 1940) and some post-hoc tests for such an evaluation.

5. Discussion

The performance of a CB algorithm in BN structure learning depends on the number of conditional independence tests and the sizes of condition sets involved in these tests. The larger the condition set, the greater the number of CI tests of high orders that have to be performed and the smaller their accuracies.

We propose the CB RAI algorithm that learns a BN structure by performing the following sequence of operations: 1) test of CI between nodes and removal of edges related to independences, 2) edge direction employing orientation rules, and 3) structure decomposition into smaller autonomous sub-structures. This sequence of operations is performed recursively for each sub-structure, along with increasing the order of the CI tests. Thereby, the RAI algorithm deals with less potential parents for the nodes on a tested edge and thus uses smaller condition sets that enable the performance of fewer CI tests of higher orders. This reduces the algorithm run-time and increases its accuracy.

By introducing orientation rules through edge direction in early stages of the algorithm and following CI tests of lower orders, the graph "backbone" is established using the most reliable CI tests. Relying on this "backbone" and its directed edges in later stages obviates the need for unnecessary CI tests and enables RAI to be less complex and sensitive to errors.

In this study, we proved the correctness of the RAI algorithm. In addition, we demonstrated empirically, using synthetically generated networks, samples of nineteen known structures, and nineteen natural databases used in classification problems, the advantage of the RAI algorithm over state-of-the-art structure learning algorithms, such as PC, TPDA, GS, GES, OR, SC and MMHC, with respect to structural correctness, number of statistical calls, run-time and classification accuracy. We note that no attempt was made to optimize the parameters of the other algorithms and the effect of such optimization was not evaluated. This is due to the fact that some of the algorithms have more than one parameter to optimize and besides, no optimization methods were proposed by the algorithm inventors. We propose such an optimization method for the RAI algorithm that uses only the training (validation) data.

We plan to extend our study in several directions. One is the comparison of RAI-based classifiers to non-BN classifiers, such as the neural network and support vector machine. Second is the incorporation of different types of prior knowledge (e.g., related to classification) into structure learning. We also intend to study error correction during learning and to allow the inclusion of hidden variables to improve representation and facilitate learning with the RAI algorithm.

Acknowledgments

The authors thank the three anonymous reviewers for their thorough reviews and helpful comments that improved the quality and clarity of the manuscript. The authors also thank the Discovery Systems Laboratory (DSL) of the Vanderbilt University, TN, for making the Causal Explorer library of algorithms and the networks tested in Aliferis et al. (2003) freely available. Special thanks due to Ms. Laura Brown of DSL for the co-operation, helpful discussions and the provision of some missing results of Aliferis et al. (2003) for comparison. This work was supported, in part, by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

References

C. F. Aliferis, I. Tsamardinos, A. Statnikov, and L. E. Brown. Causal Explorer: A causal probabilistic network learning toolkit for biomedical discovery. In *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 371–376, 2003.

S. Andreassen, F. V. Jensen, S. K. Andersen, B. Falck, U. Kjærulff, M. Woldbye, A. R. Sørensen, A. Rosenfalck, and F. Jensen. MUNIN—an expert EMG assistant. In John E. Desmedt, editor, *Computer-Aided Electromyography and Expert Systems*, chapter 21, pages 255–277. Elsevier Science Publishers, 1989.

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 *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 246–256, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

J. Cheng. PowerConstructor system. http://www.cs.ualberta.ca/~jcheng/bnpc.htm, 1998.

J. Cheng and R. Greiner. Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 101–107, 1999.

J. Cheng, D. Bell, and W. Liu. Learning Bayesian networks from data: An efficient approach based on information theory. In *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*, pages 325–331, 1997.

J. Cheng, C. Hatzis, H. Hayashi, M. Krogel, S. Morishita, D. Page, and J. Sese. KDD cup 2001 report. *ACM SIGKDD Explorations Newsletter*, 3:47–64, 2002.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

G. F. Cooper and E. A. Herskovits. Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

R. G. Cowell. Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 91–97, 2001.

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

D. Dash and M. Druzdzel. A hybrid anytime algorithm for the construction of causal models from sparse sata. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 142–149, 1999.

D. Dash and M. Druzdzel. Robust independence testing for constraint-based learning of causal structure. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 167–174, 2003.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

D. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. Technical Report R-185, Cognitive Systems Laboratory, UCLA Computer Science Department, 1992.

M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–161, 1997.

N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse-candidate" algorithm. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 206–215, 1999.

D. Grossman and P. Domingos. Learning Bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 361–368, 2004.

D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report TR-95-06, Microsoft Research, 1995.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

D. Heckerman, C. Meek, and G. F. Cooper. A Bayesian approach to causal discovery. In G. Glymour and G. Cooper, editors, *Computation, Causation and Discovery*, pages 141–165. AAAI Press, 1999.

A. Jensen and F. Jensen. MIDAS—an influence diagram for management of mildew in winter wheat. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 349–356, 1996.

R. J. Kennett, K. Korb, and A. E. Nicholson. Seebreeze prediction using Bayesian networks. In *Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 148–153, 2001.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

R. Kohavi, G. H. John, R. Long, D. Manley, and K. Pfleger. MLC++: A machine learning library in C++. In *Proceedings of the Sixth International Conference on Tools with AI*, pages 740–743, 1994.

P. Kontkanen, P. Myllymaki, T. Sliander, and H. Tirri. On supervised selection of Bayesian networks. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 334–342, 1999.

K. Kristensen and I. A. Rasmussen. The use of a Bayesian network in the design of a decision support system for growing malting barley without use of pesticides. *Computers and Electronics in Agriculture*, 33:197–217, 2002.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

P. Leray and O. François. BNT structure learning package: Documentation and experiments. Technical Report FRE CNRS 2645, Laboratoire PSI, Universitè et INSA de Rouen, 2004.

M. Marengoni, C. Jaynes, A. Hanson, and E. Riseman. Ascender II, a visual framework for 3D reconstruction. In *Proceedings of the First International Conference on Computer Vision Systems*, pages 469–488, 1999.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

C. Meek. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Carnegie Mellon University, 1997.

A. Moore and W. Wong. Optimal reinsertion: A new search operator for accelerated and more accurate Bayesian network structure learning. In *Twentieth International Conference on Machine Learning*, pages 552–559, 2003.

K. Murphy. The Bayes net toolbox for Matlab. *Computing Science and Statistics*, 33: 331–350, 2001.

D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

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

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

F. Pernkopf and J. Bilmes. Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 657–664, 2005.

T. Roos, H. Wettig, P. Grunwald, P. Myllymaki, and H. Tirri. On discriminative Bayesian network classifiers and logistic regression. *Machine Learning*, 59:267–296, 2005.

M. Singh and M. Valtorta. Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12:111–131, 1995.

P. Spirtes. An anytime algorithm for casual inference. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 213–221, 2001.

P. Spirtes and C. Meek. Learning Bayesian networks with discrete variables from data. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 294–299, 1995.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. MIT Press, 2nd edition, 2000.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006a.

I. Tsamardinos, A. Statnikov, L. E. Brown, and C. F. Aliferis. Generating realistic large Bayesian networks by tiling. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, 2006b.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

S. Yang and K. C. Chang. Comparison of score metrics for Bayesian network learning. *IEEE Transactions on Systems, Man and Cybernetics A*, 32:419–428, 2002.

R. Yehezkel and B. Lerner. Recursive autonomy identification for Bayesian network structure learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 429–436, 2005.

JMLR: Workshop and Conference Proceedings 11:171–234, 2010 Submitted 7/07; Revised 2/08; Published 1/10

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation

Constantin F. Aliferis CONSTANTIN.ALIFERIS@NYUMC.ORG
Center of Health Informatics and Bioinformatics
Department of Pathology
New York University
New York, NY 10016, USA

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Center of Health Informatics and Bioinformatics
Department of Medicine
New York University
New York, NY 10016, USA

Ioannis Tsamardinos TSAMARD@ICS.FORTH.GR
Computer Science Department, University of Crete
Institute of Computer Science, Foundation for Research and Technology, Hellas
Heraklion, Crete, GR-714 09, Greece

Subramani Mani SUBRAMANI.MANI@VANDERBILT.EDU
Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA

Xenofon D. Koutsoukos XENOFON.KOUTSOUKOS@VANDERBILT.EDU
Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37212, USA

Editor: Marina Meila

Abstract

We present an algorithmic framework for learning local causal structure around target variables of interest in the form of direct causes/effects and Markov blankets applicable to very large data sets with relatively small samples. The selected feature sets can be used for causal discovery and classification. The framework (*Generalized Local Learning*, or GLL) can be instantiated in numerous ways, giving rise to both existing state-of-the-art as well as novel algorithms. The resulting algorithms are sound under well-defined sufficient conditions. In a first set of experiments we evaluate several algorithms derived from this framework in terms of predictivity and feature set parsimony and compare to other local causal discovery methods and to state-of-the-art non-causal feature selection methods using real data. A second set of experimental evaluations compares the algorithms in terms of ability to induce local causal neighborhoods using simulated and resimulated data and examines the relation of predictivity with causal induction performance.

Our experiments demonstrate, consistently with causal feature selection theory, that local causal feature selection methods (under broad assumptions encompassing ap-

propriate family of distributions, types of classifiers, and loss functions) exhibit strong feature set parsimony, high predictivity and local causal interpretability. Although non-causal feature selection methods are often used in practice to shed light on causal relationships, we find that they cannot be interpreted causally even when they achieve excellent predictivity. Therefore we conclude that only local causal techniques should be used when insight into causal structure is sought.

In a companion paper we examine in depth the behavior of GLL algorithms, provide extensions, and show how local techniques can be used for scalable and accurate global causal graph learning.

Keywords: local causal discovery, Markov blanket induction, feature selection, classification, causal structure learning, learning of Bayesian networks

1. Introduction

This paper addresses the problem of how to learn local causal structure around a target variable of interest using observational data. We focus on two specific types of local discovery: (a) identification of variables that are direct causes or direct effects of the target, and (b) discovery of Markov blankets. A Markov Blanket of a variable T is a minimal variable subset conditioned on which all other variables are probabilistically independent of T.

Discovery of local causal relationships is significant because it plays a central role in causal discovery and classification, because of its scalability benefits, and because by naturally bridging causation with predictivity, it provides significant benefits in feature selection for classification. More specifically, solving the local causal induction problem helps understanding how natural and artificial systems work; it helps identify what interventions to pursue in order for these systems to exhibit desired behaviors; under certain assumptions, it provides minimal feature sets required for classification of a chosen response variable with maximum predictivity; and finally local causal discovery can form the basis of efficient algorithms for learning the global causal structure of all variables in the data.

The paper is organized as follows: Section 2 provides necessary background material. The section summarizes related prior work in feature selection and causal discovery; reviews recent results that connect causality with predictivity; explains the central role of local causal discovery for achieving scalable global causal induction; reviews prior methods for local causal and Markov blanket discovery and published applications; finally it introduces the open problems that are the focus of the present report. Section 3 provides formal concepts and definitions used in the paper. Section 4 provides a general algorithmic framework, *Generalized Local Learning (GLL)*, which can be instantiated in many different ways yielding sound algorithms for local causal discovery and feature selection. Section 5 evaluates a multitude of algorithmic instantiations and parameterizations from GLL and compares them to state-of-the-art local causal discovery and feature selection methods in terms of classification performance, feature set parsimony, and execution time in many real data sets. Section 6 evaluates and compares new and state-of-the-art algorithms in terms of ability to induce correct local neighborhoods using simulated data from known networks and resimulated data from real-life data sets. Section 7 discusses the experimental findings and their significance.

The experiments presented here support the conclusion that local structural learning in the form of Markov blanket and local neighborhood induction is a theoretically well-motivated and empirically robust learning framework that can serve as a powerful tool for data analysis geared toward classification and causal discovery. At the same time

several existing open problems offer possibilities for non-trivial theoretical and practical discoveries making it an exciting field of research. A companion paper (part II of the present work) studies the GLL algorithm properties empirically and theoretically, introduces algorithmic extensions, and connects local to global causal graph learning (Aliferis et al., 2010). An online supplement to the present work is available at http://www.nyuinformatics.org/downloads/supplements/JMLR2009/index.html. In addition to supplementary tables and figures, the supplement provides all software and data needed to reproduce the analyses of the present paper.

2. Background

In the present section we provide a brief review of feature selection and causal discovery research, summarize theoretical results motivating this work, present methods to speed-up scalability of discovery, give desiderata for local algorithms, review prior methods for Markov blanket and local neighborhood induction, and finally discuss open problems and focus of this paper.

2.1. Brief Review of Feature Selection and Causal Discovery Research

Variable selection for predictive modeling (also called feature selection) has received considerable attention during the last three decades both in statistics and in machine learning (Guyon and Elisseeff, 2003; Kohavi and John, 1997). Intuitively, variable selection for prediction aims to select only a subset of variables for constructing a diagnostic or predictive model for a given classification or regression task. The reasons to perform variable selection include (a) improving the model predictivity and addressing the curse-of-dimensionality, (b) reducing the cost of observing, storing, and using the predictive variables, and finally, (c) gaining an understanding of the underlying process that generates the data. The problem of variable selection is more pressing than ever, due to the recent emergence of extremely large data sets, sometimes involving tens to hundreds of thousands of variables and exhibiting a very small sample-to-variable ratio. Such data sets are common in gene expression array studies, proteomics, computational biology, text categorization, information retrieval, image classification, business data analytics, consumer profile analysis, temporal modeling, and other domains and data-mining applications.

There are many different ways to define the variable selection problem depending on the needs of the analysis. Often however, the feature selection problem for classification/prediction is defined as identifying the minimum-size subset of variables that exhibit the maximal predictive performance (Guyon and Elisseeff, 2003). Variable selection methods can be broadly categorized into *wrappers* (i.e., heuristic search in the space of all possible variable subsets using a classifier of choice to assess each subset's predictive information), or *filters* (i.e., not using the classifier per se to select features, but instead applying statistical criteria to first select features and then build the classifier with the best features). In addition, there exist learners that perform *embedded variable selection*, that is, that attempt to simultaneously maximize classification performance while minimizing the number of variables used. For example, shrinkage regression methods introduce a bias into the parameter estimation regression procedure that imposes a penalty on the size of the parameters. The parameters that are close to zero are essentially filtered-out from the predictive model.

A variety of embedded variable selection methods have been recently introduced. These methods are linked to a statement of the classification or regression problem as an optimization problem with specified loss and penalty functions. These techniques usually fall into a few broad classes: One class of methods uses the $\mathcal{L}^2$-norm penalty (also known as ridge penalty), for example, the recursive feature elimination (RFE) method is based on the $\mathcal{L}^2$-norm formulation of SVM classification problem (Rakotomamonjy, 2003; Guyon et al., 2002). Other methods are based on the $\mathcal{L}^1$-norm penalty (also known as lasso penalty), for example, feature selection via solution of the $\mathcal{L}^1$-norm formulation of SVM classification problem (Zhu et al., 2004; Fung and Mangasarian, 2004) and penalized least squares with lasso penalty on the regression coefficients (Tibshirani, 1996). A third set of methods is based on convex combinations of the $\mathcal{L}^1$- and $\mathcal{L}^2$-norm penalties, for example, feature selection using the doubly SVM formulation (Wang et al., 2006) and penalized least squares with elastic net penalty (Zou and Hastie, 2005). A fourth set uses the $\mathcal{L}^0$-norm penalty, for example, feature selection via approximate solution of the $\mathcal{L}^0$-norm formulation of SVM classification problem (Weston et al., 2003). Finally other methods use other penalties, for example, smoothly clipped absolute deviation penalty (Fan and Li, 2001).

Despite the recent emphasis on mathematically sophisticated methods such as the ones mentioned, the majority of feature selection methods in the literature and in practice are heuristic in nature in the sense that in most cases it is unknown what consists an optimal feature selection solution *independently of the class of models fitted*, and under which conditions an algorithm will output such an optimal solution.

Typical variable selection approaches also include forward, backward, forward-backward, local and stochastic search wrappers (Guyon and Elisseeff, 2003; Kohavi and John, 1997; Caruana and Freitag, 1994). The most common family of filter algorithms ranks the variables according to a score and then selects for inclusion the top k variables (Guyon and Elisseeff, 2003). The score of each variable is often the univariate (pairwise) association with the outcome variable T for different measures of associations such as the signal-to-noise ratio, the G^2 statistic and others. Information-theoretic (estimated mutual information) scores and multivariate scores, such as the weights received by a Support Vector Machine, have also been suggested (Guyon and Elisseeff, 2003; Guyon et al., 2002). Excellent recent reviews of feature selection can be found in Guyon et al. (2006a), Guyon and Elisseeff (2003) and Liu and Motoda (1998).

An emerging successful but also principled filtering approach in variable selection, and the one largely followed in this paper, is based on identifying the Markov blanket of the response ("target") variable T. The Markov blanket of T (denoted as $MB(T)$) is defined as a minimal set conditioned on which all other *measured* variables become independent of T (more details in Section 3).

While classification is often useful for *recognizing or predicting the behavior* of a system, in many problem-solving activities one needs to *change the behavior* of the system (i.e., to "manipulate it"). In such cases, knowledge of the causal relations among the various parts of the system is necessary. Indeed, in order to design new drugs and therapies, institutional policies, or economic strategies, one needs to know how the diseased organism, the institution, or the economy work. Often, heuristic methods based on multivariate or univariate associations and prediction accuracy are used to induce causation, for example, consider as causally "related" the features that have a strong association with T. Such heuristics may lead to several pitfalls and erroneous inductions, as we will show in the present paper. For principled causal discovery with known theoretical properties a causal theory is needed and classification is not, in gen-

eral, sufficient (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999). Consider the classical epidemiologic example of the tar-stained finger of the heavy smoker: it does predict important outcomes (e.g., increased likelihood for heart attack and lung cancer). However, eliminating the yellow stain by washing the finger does not alter these outcomes. While experiments can help discover causal structure, quite often experimentation is impossible, impractical, or unethical. For example, it is unethical to force people to smoke and it is currently impossible to manipulate most genes in humans in order to discover which genes cause disease and how they interact in doing so. Moreover, the discoveries anticipated due to the explosive growth of biomedical and other data cannot be made in any reasonable amount of time using solely the classical experimental approach where a single gene, protein, treatment, or intervention is attempted each time, since the space of needed experiments is immense. It is clear that computational methods are needed to catalyze the discovery process.

Fortunately, relatively recently (1980's), it was shown that it is possible to soundly infer causal relations from *observational* data in many practical cases (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999; Pearl, 1988). Since then, algorithms that infer such causal relations have been developed that can greatly reduce the number of experiments required to discover the causal structure. Several empirical studies have verified their applicability (Tsamardinos et al., 2003b; Spirtes et al., 2000; Glymour and Cooper, 1999; Aliferis and Cooper, 1994).

One of the most common methods to model and induce causal relations is by learning causal Bayesian networks (Neapolitan, 2004; Spirtes et al., 2000; Pearl, 2000). A special, important and quite broad class of such networks is the family of *faithful networks* intuitively defined as those whose probabilistic properties, and specifically the dependencies and independencies, are a direct function of their structure (Spirtes et al., 2000). Cooper and Herskovits were the first to devise a score measuring the fit of a network structure to the data based on Bayesian statistics, and used it to learn the highest score network structure (Cooper and Herskovits, 1992). Heckerman and his colleagues studied theoretically the properties of the various scoring metrics as they pertain to causal discovery (Glymour and Cooper, 1999; Heckerman, 1995; Heckerman et al., 1995). Heckerman also recently showed that Bayesian-scoring methods also assume (implicitly) faithfulness, see Chapter 4 of Glymour and Cooper (1999). Another prototypical method for learning causal relationships by inducing causal Bayesian networks is the constraint-based approach as exemplified in the PC algorithm by Spirtes et al. (2000). The PC induces causal relations by assuming faithfulness and by performing tests of independence. A network with a structure consistent with the results of the tests of independence is returned. Several other methods for learning networks have been devised subsequently (Chickering, 2003; Moore and Wong, 2003; Cheng et al., 2002a; Friedman et al., 1999b).

There may be many different networks that fit the data equally well, even in the sample limit, and that exhibit the same dependencies and independencies and are thus statistically equivalent. These networks belong to the same Markov equivalence class of causal graphs and contain the same causal edges but may disagree on the direction of some of them, that is, whether A causes B or vice-versa (Chickering, 2002; Spirtes et al., 2000). An *essential graph* is a graph where the directed edges represent the causal relations on which all equivalent networks agree upon their directionality and all the remaining edges are undirected. Causal discovery by employing causal Bayesian networks is based on the following principles. The PC (Spirtes et al., 2000), Greedy Equivalence Search (Chickering, 2003) and other prototypical or state-of-the-

art Bayesian network-learning algorithms provide theoretical guarantees, that under certain conditions such as faithfulness they will converge to a network that is statistically indistinguishable from the true, causal, data-generating network, if there is such. Thus, if the conditions hold the existence of all and the direction of some of the causal relations can be induced by these methods and graphically identified in the essential graph of the learnt network.

A typical condition of the aforementioned methods is causal sufficiency (Spirtes et al., 2000). This condition requires that for every pair of measured variables all their common direct causes are also measured. In other words, there are no hidden, unmeasured confounders for any pair of variables. Algorithms, such as the FCI, that in some cases can discover causal relationships in the presence of hidden confounding variables and selection bias, have also been designed (see Spirtes et al. 2000 and Chapter 6 of Glymour and Cooper 1999).

As it was mentioned above, using observational data alone (even a sample of an infinite size), one can infer only a Markov equivalence class of causal graphs, which may be inadequate for causal discovery. For example, it is not possible to distinguish with observational data any of these two graphs that belong to the same Markov equivalence class: $X \rightarrow Y$ and $X \leftarrow Y$. However, experimental data can distinguish between these graphs. For example, if we manipulate X and see no change in the distribution of Y, we can conclude that the data-generative graph is not $X \rightarrow Y$. This principle is exploited by active learning algorithms. Generally speaking, causal discovery with active learning can be described as follows: learn an approximation of a causal network structure from available data (which is initially only observational data), select and perform an experiment that maximizes some utility function, augment data and possibly current best causal network with the result of experiment, and repeat the above steps until some termination criterion is met.

Cooper and Yoo (1999) proposed a Bayesian scoring metric that can incorporate both observational and experimental data. Using a similar metric (Tong and Koller, 2001) designed an algorithm to select experiments that reduce the entropy of probability of alternative edge orientations. A similar but more general algorithm has been proposed in Murphy (2001) where the expected information gain of a new experiment is calculated and the experiment with the largest information gain is selected. Both above methods were designed for discrete data distributions. Pournara and Wernisch (2004) proposed another active learning algorithm that uses a loss function defined in terms of the size of transition sequence equivalence class of networks (Tian and Pearl, 2001) and can handle continuous data. Meganck et al. (2006) have introduced an active learning algorithm that is based on a general decision theoretic framework that allows to assign costs to each experiment and each measurement. It is also worthwhile to mention the GEEVE system of Yoo and Cooper (2004) that recommends which experiments to perform to discover gene-regulation pathway. This instance of causal active learning allows to incorporate preferences of the experimenter. Recent work has also provided theoretical bounds and related algorithms to minimize the number of experiments needed to infer causal structure (Eberhardt et al., 2006, 2005).

2.2. Synopsis of Theoretical Results Motivating Present Research

A key question that has been investigated in the feature selection literature is which family of methods is more advantageous: filters or wrappers. A second one is what are the "relevant" features? The latter question presumably is important because "relevant"

features should be important for discovery and so several definitions appeared defining relevancy (Guyon and Elisseeff, 2003; Kohavi and John, 1997). Finally, how can we design optimal and efficient feature selection algorithms? Fundamental theoretical results connecting Markov blanket induction for feature selection and local causal discovery to standard notions of relevance were given in Tsamardinos and Aliferis (2003). The latter paper provides a technical account and together with Spirtes et al. (2000), Pearl (2000), Kohavi and John (1997) and Pearl (1988) they constitute the core theoretical framework underpinning the present work. Here we provide a very concise description of the results in Tsamardinos and Aliferis (2003) since they partially answer these questions and pave the way to principled feature selection:

1. Relevance cannot be defined independently of the learner and the model-performance metric (e.g., the loss function used) in a way that the relevant features are the solution to the feature selection problem. The quest for a universally applicable notion of relevancy for prediction is futile.

2. Wrappers are subject to the No-Free Lunch Theorem for optimization: averaged out on all possible problems any wrapper algorithm will do as well as a random search in the space of feature subsets. Therefore, there cannot be a wrapper that is a priori more efficient than any other (i.e., without taking into account the learner and model-performance metric). The quest for a universally efficient wrapper is futile as well.

3. Any filter algorithm can be viewed as the implementation of a definition of relevancy. Because of #1, there is no filter algorithm that is universally optimal, independently of the learner and model-performance metric.

4. Because of #2, wrappers cannot guarantee universal efficiency and because of #3, filters cannot guarantee universal optimality and in that respect, neither approach is superior to the other.

5. Under the conditions that (i) the learner that constructs the classification model can actually learn the distribution $P(T \mid MB(T))$ and (ii) that the loss function is such that perfect estimation of the probability distribution of T is required with the smallest number of variables, the Markov blanket of T is the optimal solution to the feature selection problem.

6. Sound Markov blanket induction algorithms exist for faithful distributions.

7. In faithful distributions and under the conditions of #5, the strongly/weakly/irrelevant taxonomy of variables (Kohavi and John, 1997) can be mapped naturally to causal graph properties. Informally stated, strongly relevant features were defined by Kohavi and John (1997) to be features that contain information about the target not found in other variables; weakly relevant features are informative but redundant; irrelevant features are not informative (for formal definitions see Section 3). Under the causal interpretation of this taxonomy of relevancy, strongly relevant features are the members of the Markov blanket of the target variable, weakly relevant features are all variables with an undirected path to T which are not themselves members of $MB(T)$, and irrelevant features are variables with no undirected path to the target.

8. Since in faithful distributions the $MB(T)$ contains the direct causes and direct effects of T, and since state-of-the-art $MB(T)$ algorithms output the spouses separately from the direct causes and direct effects, inducing the $MB(T)$ not only solves the feature selection problem but also a form of local causal discovery problem.

Figure 1 provides a summary of the connection between causal structure and predictivity.

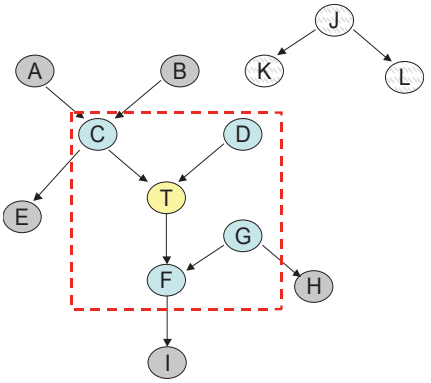


Figure 1: Relationship between causal structure and predictivity in faithful distributions. Cyan variables are members of Markov blanket of T. They are depicted inside the red dotted square (i.e., variables that have undirected path to target T and that are predictive of T given the remaining variables which makes them strongly relevant). Markov blanket variables include direct causes of T (C, D), direct effects (F), and "spouses" of T (i.e., direct causes of the direct effects of T) (G). Grey variables are non-members of Markov blanket of T that have undirected path to T. They are not predictive of T given the remaining variables but they are predictive given a subset of the remaining variables (which makes them weakly relevant). Light-gray variables are variables that do not have an undirected path to T. They are not predictive of T given any subset of the remaining variables, thus they are irrelevant.

We will refer to algorithms that perform feature selection by formal causal induction as *causal feature selection* and algorithms that do not as *non-causal*. As highly complementary to the above results we would add the arguments in favor of causal feature selection presented in Guyon et al. (2007) and recent theoretical (Hardin et al., 2004) and empirical (Statnikov et al., 2006) results that show that under the same sufficient conditions that make Markov blanket the optimal solution to the feature selection and local causal discovery problem, state-of-the-art methods such as ranking features by SVM weights (RFE being a prototypical algorithm Guyon et al. 2002) do not return the correct causal neighborhood and are not minimal, that is, do not solve the feature selection problem) even in the large sample limit.

The above theoretical results also suggest that one should not attempt to define and identify the relevant features for prediction, when discovery is the goal of the analysis.

Instead, we argue that a set of features with well-defined *causal* semantics should be identified instead: for example, the $MB(T)$, the set of direct causes and direct effects of T, the set of all (direct and indirect) causes of T, and so on.

We will investigate limitations of prominent non-causal feature selection algorithms in the companion paper (Aliferis et al., 2010).

2.3. Methods to Speed-up Discovery: Local Discovery as a Critical Tool for Scalability

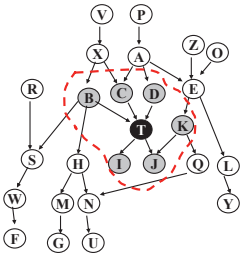
As appealing as causal discovery may be for understanding a domain, predicting effects of intervention, and pursuing principled feature selection for classification, a major problem up until recent years has been scalability. The PC algorithm is worst-case exponential (Spirtes et al., 2000) and in practical settings it cannot typically handle more than a hundred variables. The FCI algorithm is similarly worst-case intractable (Spirtes et al., 2000) and does not handle more than a couple of dozen of variables practically. Learning Bayesian networks with Bayesian scoring techniques is NP-Hard (Chickering et al., 1994). Heuristic hill-climbing techniques such as the Sparse Candidate Algorithm (Friedman et al., 1999b) do not provide guaranteed correct solutions, neither they are very efficient (they can cope with a few hundred variables at the most in practical applications).

With the advent of massive data sets in biology, medicine, information retrieval, the WWW, finance, economics, and so on, scalability has become a critical requirement for practical algorithms. In early 2000's predictions about the feasibility of causal discovery in high-dimensional data were bleak (Silverstein et al., 2000). A variety of methods to scale up causal discovery have been devised to address the problem:

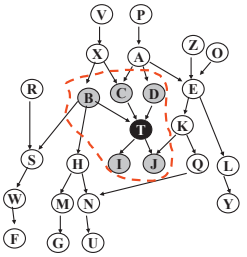
1. Learn the full graph but focus on special types of distributions;
2. Exploit domain knowledge to speed-up learning;
3. Abandon the effort to learn the full causal graph and instead develop methods that find a portion of the true arcs (not specific to some target variable);
4. Abandon the effort to learn the full causal graph and instead develop methods that learn the local neighborhood of a specific target variable directly;
5. Abandon the effort to learn the fully oriented causal graph and instead develop methods that learn the unoriented graph;
6. Induce constrains of the possible relationships among variables and then learn the full causal graph.

Techniques #1 and #2 were introduced in Chow and Liu (1968) for learning tree-like graphs and Naïve-Bayes graphs (Duda and Hart, 1973), while modern versions are exemplified in (i) TAN/BAN classifiers that relax the Naïve-Bayes structure (Cheng and Greiner, 2001, 1999; Friedman et al., 1997), (ii) efficient complete model averaging of Naïve-Bayes classifiers (Dash and Cooper, 2002), and (iii) algorithm TPDA which restricts the class of distributions so that learning becomes from worst-case intractable to solvable in 4^{th} degree polynomial time to the number of variables (and quadratic if prior knowledge about the ordering of variables is known) (Cheng et al., 2002a). Technique #3 was introduced by Cooper (1997) and replaced learning the complete graph by learning only a small portion of the edges (not pre-specified by the user but determined by the

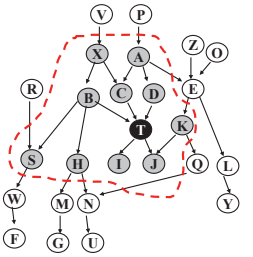
discovery method). Techniques #4 – 6 pertain to local learning: Technique #4 seeks to learn the complete causal neighbourhood around a target variable provided by the user (Aliferis et al., 2003a; Tsamardinos et al., 2003b). We emphasize that local learning (technique #4) is not the same as technique #3 (incomplete learning) although inventors of incomplete methods often call them 'local'. Technique #5 abandons directionality and learns only a fully connected but undirected graph by using local learning methods (Tsamardinos et al., 2006; Brown et al., 2005). Often post-processing with additional algorithms can provide directionality. The latter can also be obtained by domain-specific criteria or experimentation. Finally, technique #6 uses local learning to restrict the search space for full-graph induction algorithms (Tsamardinos et al., 2006; Aliferis and Tsamardinos, 2002b).



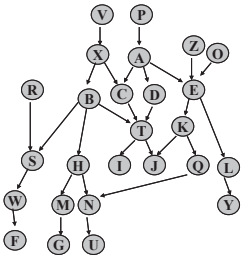
Problem #1: Consider a target variable T and discover Markov Blanket of T.



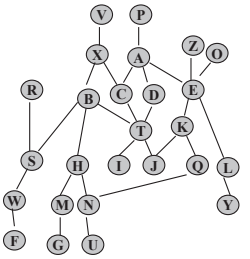
Problem #2: Consider a target variable T and discover Parents and Children of T.



Problem #3: Consider a target variable T and discover regions (e.g., of depth 2 edges) around T.



Problem #4: Discover directed graph.



Problem #5: Discover undirected graph.

Figure 2: Five types of causal discovery from local (types 1, 2), to global (4, 5) and intermediate (3). Specialized algorithms that solve type 2 (local causes and effects) can become building blocks for relatively efficiently solving all other types of causal discovery as well (see text for details).

In the present paper we explore methods to learn local causal neighborhoods and test them in high-dimensional data sets. In the companion paper (Aliferis et al., 2010) we provide a framework for building global graphs using the local methods. Incomplete learning (technique #3) is not pursued because it is redundant in light of the other (complete) local and global learning approaches. Figure 2 provides a visual reference guide to the kinds of causal discovery problems the methods in the present work are able to address by starting from local causal discovery.

2.4. Desiderata for Local Algorithms, Brief Review of Prior Methods for Markov Blanket and Local Neighborhood Induction

An ideal local learning algorithm should have three characteristics: (a) well-defined properties, especially broadly applicable conditions that guarantee correctness, (b) good performance in practical distributions and corresponding data sets, including ones with small sample and many features, and finally (c) scalability in terms of running time. We briefly review progress made in the field toward these goals.

Firm theoretical foundations of Bayesian networks were laid down by Pearl and his co-authors (Pearl, 1988). Furthermore, all local learning methods exploit either the constraint-based framework for causal discovery developed by Spirtes, Glymour, Schienes, Pearl, and Verma and their co-authors (Spirtes et al., 2000; Pearl, 2000; Pearl and Verma, 1991) or the Bayesian search-and-score Bayesian network learning framework introduced by Cooper and Herskovits (1992). The relevant key contributions were covered in Section 2.1 and will not be repeated here.

While the above foundations were introduced and developed in the span of at least the last 30 years, local learning is no more than 10 years old. Specialized Markov blanket learning methods were first introduced in 1996 (Koller and Sahami, 1996), incomplete causal methods in 1997 (Cooper, 1997), and local causal discovery methods (for targeted complete induction of direct causes and effects) were first introduced in 2002 and 2003 (Tsamardinos et al., 2003b; Aliferis and Tsamardinos, 2002a). In 1996, Koller et al. introduced a heuristic algorithm for inducing the Markov blanket from data and tested the algorithm in simulated, real text, and other types of data from the UCI repository (Koller and Sahami, 1996). In 1997 Cooper and colleagues introduced and applied the heuristic method K2MB for finding the Markov blanket of a target variable in the task of predicting pneumonia mortality (Cooper, 1997). In 1997 Cooper introduced an incomplete method for causal discovery (Cooper et al., 1997). The algorithm was able to circumvent lack of scalability of global methods by returning a subset of arcs from the full network. To avoid notational confusion we point out that the algorithm was termed LCD (local causal discovery) despite being an *incomplete rather than local* algorithm as local algorithms are defined in the present paper (i.e., focused on some user-specified target variable or localized region of the network). A revision of the algorithm termed LCD2 was presented in Mani and Cooper (1999).

In 1999 Margaritis and Thrun introduced the GS algorithm with the intent to induce the Markov blanket for the purpose of speeding up global network learning (i.e., not for feature selection) (Margaritis and Thrun, 1999). GS was the first published sound Markov blanket induction algorithm. The weak heuristic used by GS combined with the need to condition on at least as many variables simultaneously as the Markov blanket size makes it impractical for many typical data sets since the required sample grows exponentially to the size of the Markov blanket. This in turn forces the algorithm to stop its execution prematurely (before it identifies the complete Markov blanket) because it cannot grow the conditioning set while performing reliable tests of independence. Evaluations of GS by its inventors were performed in data sets with a few dozen variables leaving the potential of scalability largely unexplored.

In 2001 Cheng et al. applied the TPDA algorithm (a global BN learner) (Cheng et al., 2002a) to learn the Markov blanket of the target variable in the Thrombin data set in order to solve a prediction problem of drug effectiveness on the basis of molecular characteristics (Cheng et al., 2002b). Because TPDA could not be run with more than a few hundred variables efficiently, they pre-selected 200 variables (out of 139,351 total) using univariate filtering. Although this procedure in general will not find the true

Markov blanket (because otherwise-unconnected with the target spouses can be missed, many true parents and children may not be in the first 200 variables, and many non-Markov blanket members cannot be eliminated), the resulting classifier performed very well winning the 2001 KDD Cup competition.

Friedman et al. proposed a simple Bootstrap procedure for determining membership in the Markov blanket for small sample situations (Friedman et al., 1999a). The Markov blanket in this method is to be extracted from the full Bayesian network learned by the SCA (Sparse Candidate Algorithm) learner (Friedman et al., 1999b).

In 2002 and 2003 Tsamardinos, Aliferis, et al. presented a modified version of GS, termed IAMB and several variants of the latter that through use of a better inclusion heuristic than GS and optional post-processing of the tentative and final output of the local algorithm with global learners would achieve true scalability to data sets with many thousands of variables and applicability in modest (but not very small) samples (Tsamardinos et al., 2003a; Aliferis et al., 2002). IAMB and several variants were tested both in the high-dimensional Thrombin data set (Aliferis et al., 2002) and in data sets simulated from both existing and random Bayesian networks (Tsamardinos et al., 2003a). The former study found that IAMB scales to high-dimensional data sets. The latter study compared IAMB and its variants to GS, Koller-Sahami, and PC and concluded that IAMB variants on average perform best in the data sets tested.

In 2003 Tsamardinos and Aliferis presented a full theoretical analysis explaining relevance as defined by Kohavi and John (1997) in terms of Markov blanket and causal connectivity (Tsamardinos and Aliferis, 2003). They also provided theoretical results about the strengths and weaknesses of filter versus wrapper algorithms, the impossibility of a universal definition of relevance, and the optimality of Markov blanket as a solution to the feature selection problem in formal terms. These results were summarized in Section 2.2.

The extension of Sparse Candidate Algorithm to create a local-to-global learning strategy was first introduced in Aliferis and Tsamardinos (2002b) and led to the MMHC algorithm introduced and evaluated in Tsamardinos et al. (2006). MMHC was shown in Tsamardinos et al. (2006) to achieve best-of-class performance in quality and scalability compared to most state-of-the-art global network learning algorithms.

In 2002 Aliferis et al. also introduced parallel and distributed versions of the IAMB family of algorithms (Aliferis et al., 2002). These serve as the precursor of the parallel and distributed local neighborhood learning method presented in the companion paper (Aliferis et al., 2010). The precursor of the GLL framework was also introduced by Aliferis and Tsamardinos in 2002 for the explicit purpose of reducing the sample size requirements of IAMB-style algorithms (Aliferis and Tsamardinos, 2002a).

In 2003 Aliferis et al. introduced algorithm HITON[1] Aliferis et al., and Tsamardinos et al. introduced algorithms MMPC and MMMB (Aliferis et al., 2003a; Tsamardinos et al., 2003b). These are the first concrete algorithms that would find sets of direct causes or direct effects and Markov blankets in a scalable and efficient manner. HITON was tested in 5 biomedical data sets spanning clinical, text, genomic, structural and proteomic data and compared against several feature selection methods with excellent results in parsimony and classification accuracy (Aliferis et al., 2003a). MMPC was tested in data simulated from human-derived Bayesian networks with excellent results in quality and scalability. MMMB was tested in the same data sets and compared to prior algorithms such as Koller-Sahami algorithm and IAMB variants with superior results in the quality of Markov blankets. These benchmarking and comparative evaluation experiments

1. From the Greek word "Χιτών" meaning "cloak", and pronounced <hee tó n>.

provided evidence that the local learning approach held not only theoretical but also practical potential.

HITON-PC, HITON-MB, MMPC, and MMMB algorithms lacked so-called "symmetry correction" (Tsamardinos et al., 2006), however HITON used a wrapping post-processing that at least in principle removed this type of false positives. The symmetry correction was introduced in 2005 and 2006 by Tsamardinos et al. in the context of the introduction of MMHC (Tsamardinos et al., 2006, 2005). Peña et al. also published work pointing to the need for a symmetry correction in MMPC (Peña et al., 2005b).

HITON was applied in 2005 to understand physician decisions and guideline compliance in the diagnosis of melanomas (Sboner and Aliferis, 2005). HITON has been applied for the discovery of biomarkers in human cancer data using microarrays and mass spectrometry and is also implemented in the GEMS and FAST-AIMS systems for the automated analysis of microarray and mass spectrometry data respectively (Statnikov et al., 2005b; Fananapazir et al., 2005). In a recent extensive comparison of biomarker selection algorithms (Aliferis et al., 2006a,b) it was found that HITON outperforms 16 state-of-the-art representatives from all major biomarker algorithmic families in terms of combined classification performance and feature set parsimony. This evaluation used 9 human cancer data sets (gene expression microarray and mass spectrometry) in 10 diagnostic and outcome (i.e., survival) prediction classification tasks. In addition to the above real data, resimulation was also used to create two gold standard network structures, one re-engineered from human lung cancer data and one from yeast data. Several applications of HITON in text categorization have been published where the algorithm was used to understand complex "black box" SVM models and convert complex models to Boolean queries usable by Boolean interfaces of Medline (Aphinyanaphongs and Aliferis, 2004), to examine the consistency of editorial policies in published journals (Aphinyanaphongs et al., 2006), and to predict drug-drug interactions (Duda et al., 2005). HITON was also compared with excellent results to manual and machine feature selection in the domain of early graft failure in patients with liver transplantations (Hoot et al., 2005).

In 2003 Frey et al. explored the idea of using decision tree induction to indirectly approximate the Markov blanket (Frey et al., 2003). They produced promising results, however a main problem with the method was that it requires a threshold parameter that cannot be optimized easily. Furthermore, as we show in the companion paper (Aliferis et al., 2010) decision tree induction is subject to synthesis and does not select only the Markov blanket members.

In 2004 Mani et al. introduced BLCD-MB, which resembles IAMB but using a Bayesian scoring metric rather than conditional independence testing (Mani and Cooper, 2004). The algorithm was applied with promising results in infant mortality data (Mani and Cooper, 2004).

A method for learning regions around target variables by recursive application of MMPC or other local learning methods was introduced in Tsamardinos et al. (2003c). Peña et al. applied interleaved MMPC for learning regions in the domain of bioinformatics (Peña et al., 2005a).

In 2006 Gevaert et al. applied K2MB for the purpose of learning classifiers that could be used for prognosis of breast cancer from microarray and clinical data (Gevaert et al., 2006) . Univariate filtering was used to select 232 genes before applying K2MB.

Other recent efforts in learning Markov blankets include the following algorithms: PCX, which post-processes the output of PC (Bai et al., 2004); KIAMB, which addresses some violations of faithfulness using a stochastic extension to IAMB (Peña et al., 2007);

FAST-IAMB, which speeds up IAMB (Yaramakala and Margaritis, 2005); and MBFS, which is a PC-style algorithm that returns a graph over Markov blanket members (Ramsey, 2006).

2.5. Open Problems and Focus of Paper

The focus of the present paper is to describe state-of-the-art algorithms for inducing direct causes and effects of a response variable or its Markov blanket using a novel cohesive framework that can help in the analysis, understanding, improvement, application (including configuration / parameterization) and dissemination of the algorithms. We furthermore study comparative performance in terms of predictivity and parsimony of state-of-the-art local causal algorithms; we compare them to non-causal algorithms in real and simulated data sets using the same criteria; and show how novel algorithms can be obtained. A second major hypothesis (and set of experiments in the present paper) is that non-causal feature selection methods may yield predictively optimal feature sets while from a causal perspective their output is unreliable. Testing this hypothesis has tremendous implications in many areas (e.g., analysis of biomedical molecular data) where highly predictive variables (biomarkers) of phenotype (e.g., disease or clinical outcome) are often interpreted as being causally implicated for the phenotype and great resources are invested in pursuing these markers for new drug development and other research.

In the second part of our work (Aliferis et al., 2010) we address gaps in the theoretical understanding of local causal discovery algorithms and provide empirical and theoretical analyses of their behavior as well as several extensions including algorithms for learning the full causal graph using a divide-and-conquer local learning approach.

3. Notation and Definitions

In the present paper we use Bayesian networks as the language in which to represent data generating processes and causal relationships. We thus first formally define causal Bayesian networks. Recall that in a directed acyclic graph (DAG), a node A is the parent of B (B is the child of A) if there is a direct edge from A to B, A is the ancestor of B (B is the descendant of A) if there is a direct path from A to B. "Nodes", "features", and "variables" will be used interchangeably.

3.1. Notation

We will denote variables with uppercase letters X, Y, Z, values with lowercase letters, x, y, z, and sets of variables or values with boldface uppercase or lowercase respectively. A "target" (i.e., response) variable is denoted as T unless stated otherwise.

Definition 1 ***Conditional Independence****. Two variables X and Y are conditionally independent given* $\mathbf{Z}$*, denoted as* $I(X, Y \mid \mathbf{Z})$*, iff* $P(X = x, Y = y \mid \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z})P(Y = y \mid \mathbf{Z} = \mathbf{z})$*, for all values* $x, y, \mathbf{z}$ *of* $X, Y, \mathbf{Z}$ *respectively, such that* $P(\mathbf{Z} = \mathbf{z}) > 0$*.*

Definition 2 ***Bayesian network*** $\langle \mathbf{V}, G, J \rangle$*. Let* $\mathbf{V}$ *be a set of variables and J be a joint probability distribution over all possible instantiations of* $\mathbf{V}$*. Let G be a directed acyclic graph (DAG) such that all nodes of G correspond one-to-one to members of* $\mathbf{V}$*. We require that for every node* $A \in \mathbf{V}$*, A is probabilistically independent of all non-descendants of A, given the parents of A (i.e.,* Markov Condition *holds). Then we call the triplet* $\langle \mathbf{V}, G, J \rangle$ *a Bayesian network*

(abbreviated as "BN"), or equivalently a belief network or probabilistic network (Neapolitan, 1990).

Definition 3 ***Operational criterion for causation.*** *Assume that a variable A can be forced by a hypothetical experimenter to take values a_i. If the experimenter assigns values to A according to a uniformly random distribution over values of A, and then observes $P(B \mid A = a_i) \neq P(B \mid A = a_j)$ for some i and j, (and within a time window dt), then variable A is a cause of variable B (within dt).*

We note that randomization of values of A serves to eliminate any combined causative influences on both A and B. We also note that universally acceptable definitions of causation have eluded scientists and philosophers for centuries. Indeed the provided criterion is not a proper definition, because it examines one cause at a time (thus multiple causation can be missed), it assumes that a hypothetical experiment is feasible even when in practice this is not attainable, and the notion of "forcing" variables to take values presupposes a special kind of causative primitive that is formally undefined. Despite these limitations, the above criterion closely matches the notion of a Randomized Controlled Experiment which is a de facto standard for causation in many fields of science, and following common practice in the field (Glymour and Cooper, 1999) will serve operationally the purposes of the present paper.

Definition 4 ***Direct and indirect causation.*** *Assume that a variable A is a cause of variable B according to the operational criterion for causation in definition 3. A is an indirect cause for B with respect to a set of variables $\boldsymbol{V}$, iff A is not a cause of B for some instantiation of values of $\boldsymbol{V} \setminus \{A, B\}$, otherwise A is a direct cause of B.*

Definition 5 ***Causal probabilistic network (a.k.a. causal Bayesian network).*** *A causal probabilistic network (abbreviated as "CPN") $\langle \boldsymbol{V}, G, J \rangle$ is the Bayesian network $\langle \boldsymbol{V}, G, J \rangle$ with the additional semantics that if there is an edge $A \rightarrow B$ in G then A directly causes B (for all $A, B \in \boldsymbol{V}$) (Spirtes et al., 2000).*

Definition 6 ***Faithfulness.*** *A directed acyclic graph G is faithful to a joint probability distribution J over variable set $\boldsymbol{V}$ iff every independence present in J is entailed by G and the Markov Condition. A distribution J is faithful iff there exists a directed acyclic graph G such that G is faithful to J (Spirtes et al., 2000; Glymour and Cooper, 1999).*

It follows from the Markov Condition that in a CPN $C = \langle \boldsymbol{V}, G, J \rangle$ every conditional independence entailed by the graph G is also present in the probability distribution J encoded by C. *Thus, together faithfulness and the causal Markov Condition establish a close relationship between a causal graph G and some empirical or theoretical probability distribution J. Hence we can associate statistical properties of the sample data with causal properties of the graph of the CPN.* The d-separation criterion determines all independencies entailed by the Markov Condition and a graph G.

Definition 7 ***d-separation, d-connection.*** *A* collider *on a path p is a node with two incoming edges that belong to p. A path between X and Y given a conditioning set $\boldsymbol{Z}$ is open, if (i) every collider of p is in $\boldsymbol{Z}$ or has a descendant in $\boldsymbol{Z}$, and (ii) no other nodes on p are in $\boldsymbol{Z}$. If a path is not open, then it is* blocked. *Two variables X and Y are d-separated given a conditioning set $\boldsymbol{Z}$ in a BN or CPN C iff every path between X, Y is blocked (Pearl, 1988).*

Property 1 *Two variables X and Y are d-separated given a conditioning set* $\boldsymbol{Z}$ *in a faithful BN or CPN iff* $I(X, Y \mid \boldsymbol{Z})$ *(Spirtes et al., 2000). It follows, that if they are d-connected, they are conditionally dependent.*

Thus, in a faithful CPN, *d*-separation captures *all* conditional dependence and independence relations that are encoded in the graph.

Definition 8 ***Markov blanket of*** T*, denoted as* $MB(T)$*. A set* $MB(T)$ *is a minimal set of features with the following property: for every variable subset* $\boldsymbol{S}$ *with no variables in* $MB(T)$*,* $I(\boldsymbol{S}, T \mid MB(T))$*. In Pearl's terminology this is called the Markov Boundary (Pearl, 1988).*

Property 2 *The* $MB(T)$ *of any variable T in a faithful BN or a CPN is unique (Tsamardinos et al., 2003b) (also directly derived from Pearl and Verma 1991 and Pearl and Verma 1990).*

Property 3 *The* $MB(T)$ *in a faithful CPN is the set of parents, children, and parents of children (i.e., "spouses") of T (Pearl, 2000, 1988).*

Definition 9 ***Causal sufficiency****. For every pair of measured variables, all their common causes are also measured.*

Definition 10 ***Feature selection problem****. Given a sample S of instantiations of variable set* $\boldsymbol{V}$ *drawn from distribution D, a classifier induction algorithm C and a loss function L, find: smallest subset of variables* $\boldsymbol{F} \subseteq \boldsymbol{V}$ *such that* $\boldsymbol{F}$ *minimizes expected loss* $L(M, D)$ *in distribution D where M is the classifier model (induced by C from sample S projected on* $\boldsymbol{F}$*).*

In the above definition, we mean "exact" minimization of $L(M, D)$. In other words, out of all possible subsets of variable set $\boldsymbol{V}$, we are interested in subsets $\boldsymbol{F} \subseteq \boldsymbol{V}$ that satisfy the following two criteria: (i) $\boldsymbol{F}$ minimizes $L(M, D)$ and (ii) there is no subset $\boldsymbol{F}^* \subseteq \boldsymbol{V}$ such that $|\boldsymbol{F}^*| < |\boldsymbol{F}|$ and $\boldsymbol{F}^*$ also minimizes $L(M, D)$.

Definition 11 ***Wrapper feature selection algorithm****. An algorithm that tries to solve the Feature Selection problem by searching in the space of feature subsets and evaluating each one with a user-specified classifier and loss function estimator.*

Definition 12 ***Filter feature selection algorithm****. An algorithm designed to solve the Feature Selection problem by looking at properties of the data and not by applying a classifier to estimate expected loss for different feature subsets.*

Definition 13 ***Causal feature selection algorithm****. An algorithm designed to solve the Feature Selection problem by (directly or indirectly) inducing causal structure and by exploiting formal connections between causation and predictivity.*

Definition 14 ***Non-causal feature selection algorithm****. An algorithm that tries to solve the Feature Selection problem without reference to the causal structure that underlies the data.*

Definition 15 ***Irrelevant, strongly relevant, weakly relevant, relevant feature (with respect to target variable*** T***)****. A variable set* $\boldsymbol{I}$ *that conditioned on every subset of the remaining variables does not carry predictive information about T is irrelevant to T. Variables that are not irrelevant are called relevant. Relevant variables are strongly relevant if they are predictive for T given the remaining variables, while a variable is weakly relevant if it is non-predictive for T given the remaining variables (i.e., it is not strongly relevant) but it is predictive given some subset of the remaining variables.*

4. A General Framework for Local Learning

In this section we present a formal general framework for learning local causal structure. Such a framework enables a systematic exploration of a family of related but not identical algorithms which can be seen as instantiations of the same broad algorithmic principles encapsulated in the framework. Also, the framework allows us to think about formal conditions for correctness not only at the algorithm level but also at the level of algorithm family. We are thus able to identify two distinct sets of assumptions for correctness: the more general set of assumptions (*admissibility rules*) applies to the generative algorithms and provides a set of flexible rules for constructing numerous algorithmic instantiations each one of which is guaranteed to be correct provided that in addition a more specific and fixed set of assumptions hold (i.e., specific sufficient conditions for correctness of the algorithms that are instantiations of the generative framework).

We consider the following two problems of local learning:

Problem 1 *Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the direct causes and direct effects of T.*

Problem 2 *Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the direct causes, direct effects, and the direct causes of the direct effects of T.*

From the work of Spirtes et al. (2000) and Pearl (2000, 1988) we know that when the data are observational, causal sufficiency holds for the variables $\boldsymbol{V}$, and the distribution P is faithful to a causal Bayesian network, then the direct causes, direct effects, and direct causes of the direct effects of T, correspond to the parents, children, and spouses of T respectively in that network.

Thus, in the context of the above assumptions, Problem 1 seeks to identify the parents and children set of T in a Bayesian network G faithful to P; we will denote this subset as $PC_G(T)$. There may be several networks that faithfully capture distribution P, however, as we have shown in Tsamardinos et al. (2003b) (also directly derived from Pearl and Verma 1991, 1990) $PC_G(T) = PC_{G'}(T)$, for any two networks G and G' faithful to the same distribution. So, the set of parents and children of T is unique among all Bayesian networks faithful to the same distribution and so we will drop the superscript and denote it simply as $PC(T)$. Notice that, a node may be a parent of T in one network and a child of T in another, for example, the graphs $X \leftarrow T$ and $X \rightarrow T$ may both be faithful to the same distribution. However, the set of parents and children of T, that is, $\{X\}$, remains the same in both networks. Finally, by Theorem 4 in Tsamardinos et al. (2003b) we know that the Markov blanket $MB(T)$ is unique in all networks faithful to the same distribution. Therefore, under the assumptions of the existence of a causal Bayesian network that faithfully captures P and causal sufficiency of $\boldsymbol{V}$, the problems above can be recast as follows:

Problem 3 *Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the $PC(T)$.*

Problem 4 *Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the $MB(T)$.*

Problem 1 is geared toward local causal discovery, while Problem 2 is oriented toward causal feature selection for classification. The solutions to these problems can form the basis for solving several other related local discovery problems, such as learning the unoriented set of causal relations (skeleton of a Bayesian network), a region of interest of a given depth of d edges around T, or further analyze the data to discover the orientation of the causal relations.

The *Generalized Local Learning* (GLL) framework consists of two main types of algorithms: GLL-PC (GLL Parent and Children) for Problem 1 and GLL-MB for Problem 2.

4.1. Discovery of the $PC(T)$ Set

Identification of the $PC(T)$ set is based on the following theorem in Spirtes et al. (2000):

Theorem 1 *In a faithful BN $\langle \boldsymbol{V}, G, P \rangle$ there is an edge between the pair of nodes $X \in \boldsymbol{V}$ and $Y \in \boldsymbol{V}$ iff $\neg I(X, Y \mid \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$.*

Any variable X that does have an edge with T belongs to the $PC(T)$. Thus, the theorem gives rise to an immediate algorithm for identifying $PC(T)$: for any variable $X \in \boldsymbol{V} \setminus \{T\}$, and all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$, test whether $\mathrm{I}(X, T \mid \boldsymbol{Z})$. If such a $\boldsymbol{Z}$ exists for which $\mathrm{I}(X, T \mid \boldsymbol{Z})$, then $X \notin PC(T)$, otherwise $X \in PC(T)$. This algorithm is equivalent to a "localized version" of SGS (Spirtes et al., 2000). The problem of course is that the algorithm is very inefficient because it tests all subsets of the variables and thus does not scale beyond problems of trivial size. The order of complexity is $\mathrm{O}(|\boldsymbol{V}|2^{|\boldsymbol{V}|-2})$. The general framework presented below attempts to characterize not only the above algorithm but also efficient implementations of the theorem that maintain soundness.

There are several observations that lead to more efficient but still sound algorithms. First notice that, once a subset $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ has been found s.t. $\mathrm{I}(X, T \mid \boldsymbol{Z})$ there is no need to perform any other test of the form $\mathrm{I}(X, T \mid \boldsymbol{Z}')$: we know that $X \notin PC(T)$. Thus, the sooner we identify good candidate subsets $\boldsymbol{Z}$ that can render the variables conditionally independent from T, the fewer tests will be necessary.

Second, to determine whether $X \in PC(T)$ there is no need to test whether $\neg\mathrm{I}(X, T \mid \boldsymbol{Z})$ for all subsets $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ but only for all subsets $\boldsymbol{Z}' \subseteq Parents_G(T) \setminus \{X\}$ and all $\boldsymbol{Z}' \subseteq Parents_G(X) \setminus \{T\}$ where G is any network faithful to the distribution. To see this, let us first assume that there is no edge between X and T. Notice that either X is a non-descendant of T or T is a non-descendant of X since the network is acyclic and they cannot be both descendants of each other. If X is a non-descendant of T in G, then by the Markov Condition we know that there is a subset $\boldsymbol{Z}$ of $Parents_G(T) = Parents_G(T) \setminus \{X\}$ (the equality because we assume no edge between T and X) such that $\mathrm{I}(X, T \mid \boldsymbol{Z})$. Similarly, if T is a non-descendant of X in G then there is $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$ such that $\mathrm{I}(X, T \mid \boldsymbol{Z})$. Conversely, if there is an edge $X \to T$ or $T \to X$, then the dependence $\neg\mathrm{I}(X, T \mid \boldsymbol{Z})$ holds for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (by the theorem), thus also holds for all $\boldsymbol{Z} \subseteq Parents_G(T) \setminus \{X\}$ or $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$. We just proved that:

Proposition 1 *In a faithful BN $\langle \boldsymbol{V}, G, P \rangle$ there is an edge between the pair of nodes $X \in \boldsymbol{V}$ and $T \in \boldsymbol{V}$ iff $\neg I(X, T \mid \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$ and $\boldsymbol{Z} \subseteq Parents_G(T) \setminus \{X\}$.*

Since the networks in most practical problems are relatively sparse, if we knew the sets
$Parents_G(T)$ and $Parents_G(X)$ then the number of subsets that would need to be checked for conditional independence for each $X \in PC(T)$ is significantly smaller:

$|2^{|\boldsymbol{V}\setminus\{T,X\}|} \gg |2^{|Parents_G(X)|}| + |2^{|Parents_G(T)|}|$. Of course, we do not know the sets $Parents_G(T)$ and $Parents_G(X)$ but one could work with any superset of them as shown by the following proposition:

Proposition 2 *In a faithful BN* $\langle \boldsymbol{V}, G, P\rangle$ *there is an edge between the pair of nodes* $X \in \boldsymbol{V}$ *and* $T \in \boldsymbol{V}$ *iff* $\neg I(X, T \mid \boldsymbol{Z})$, *for all* $\boldsymbol{Z} \subseteq \boldsymbol{S}$ *and* $\boldsymbol{Z} \subseteq \boldsymbol{S}'$, *where* $Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X, T\}$ *and* $Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \{X, T\}$.

Proof If there is an edge between the pair of nodes X and T then $\neg\mathrm{I}(X, T \mid \boldsymbol{Z})$, for all subsets $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (by Theorem 1) and so $\neg\mathrm{I}(X, T \mid \boldsymbol{Z})$ for all $\boldsymbol{Z} \subseteq \boldsymbol{S}$ and $\boldsymbol{Z} \subseteq \boldsymbol{S}'$ too. Conversely, if there is no edge between the pair of nodes X and T, then $\mathrm{I}(X, T \mid \boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq Parents_G(X) = Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S}$ or $\boldsymbol{Z} \subseteq Parents_G(T) = Parents_G(T) \setminus \{X\} \subseteq \boldsymbol{S}'$ (by Proposition 1). ■

Now, the sets $Parents_G(X)$ and $Parents_G(T)$ depend on the specific network G that we are trying to learn. As we mentioned however, there may be several such statistically equivalent networks among which we cannot differentiate from the data, forming an equivalence class. Thus, it is preferable to work with supersets of $Parents_G(T)$ and $Parents_G(X)$ that do not depend on a specific network member of the class: these supersets are the sets $PC(T)$ and $PC(X)$.

Let us suppose that we have available a superset of $PC(T)$ called $TPC(T)$ (tentative PC). For any node $X \in TPC(T)$ if $\mathrm{I}(X, T \mid \boldsymbol{Z})$ for some $\boldsymbol{Z} \subseteq TPC(T) \setminus \{X, T\}$, then by Proposition 2, we know that X has no edge with T, that is, $X \notin PC(T)$. So, X should also be removed from $TPC(T)$ to obtain a better approximation of $PC(T)$. If however, $\neg\mathrm{I}(X, T \mid \boldsymbol{Z})$ for all $\boldsymbol{Z} \subseteq TPC(T) \setminus \{X, T\}$, then it is still possible that $X \notin PC(T)$ because there may be a set $\boldsymbol{Z} \subseteq PC(X)$ where $\boldsymbol{Z} \nsubseteq PC(T)$ for which $\mathrm{I}(X, T \mid \boldsymbol{Z})$.

Is there actually a case, where X cannot be made independent of T by conditioning on some subset of $PC(T)$? We know that all non-descendants of T can be made independent of T conditioned on a subset of its parents, thus, if there is such an X it has to be a descendant of T. Figure 3 shows such a case. These situations are rare in practice as indicated by our empirical results in Sections 5 and 6, which implies that by conditioning on all subsets of $TPC(T)$ one will approximate $PC(T)$ quite closely.

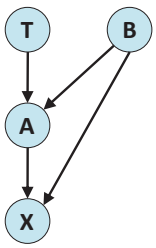


Figure 3: $PC(T) = \{A\}, PC(X) = \{A, B\}, X \notin PC(T)$. Notice that, there is no subset of $PC(T)$ that makes T conditionally independent of X : $\neg I(X, T \mid \emptyset), \neg I(X, T \mid A)$. However, there is a subset of $PC(X)$ for which X and T become conditionally independent: $I(X, T \mid \{A, B\})$. The Extended $PC(T)$ (see Definition 16 in this section) is $EPC(T) = \{A, X\}$.

Definition 16 *We call the Extended $PC(T)$, denoted as $EPC(T)$, the set $PC(T)$ union the set of variables X for which $\neg I(X, T \mid \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$.*

The previous results allow us to start building algorithms that operate locally around T in order to find $PC(T)$ efficiently and soundly. Consider first the sketch of the algorithm below:

1: Find a superset $TPC(T)$ of $PC(T)$
2: **for** each variable $X \in TPC(T)$ **do**
3: **if** $\exists \boldsymbol{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. $I(X, T \mid \boldsymbol{Z})$ **then**
4: remove X from $TPC(T)$
5: **end if**
6: **end for**
7: Return $TPC(T)$

This algorithm will output $TPC(T) \subseteq EPC(T)$. To ensure we end up with the exact $PC(T)$ we can use the following pruning algorithm:

1: **for all** $X \in TPC(T)$ **do** {returned from Algorithm 4.1}
2: **if** $T \notin TPC(X)$ **then**
3: remove X from $TPC(T)$ {$TPC(X)$ is obtained by running Algorithm 4.1}
4: **end if**
5: **end for**

In essence, the second algorithm checks for every $X \in TPC(T)$ whether the *symmetrical relation* holds: $T \in TPC(X)$. If the symmetry is broken, we know that $X \notin PC(T)$ since the parents-and-children relation is symmetrical.

What is the complexity of the above algorithms? In Algorithm 4.1 if step 1 is performed by an Oracle with constant cost, and with $TPC(T)$ equal to $PC(T)$, then the first algorithm requires an order of $O(|\boldsymbol{V}|2^{|PC(T)|})$ tests. The second algorithm will require an order of $O(|\boldsymbol{V}|2^{|PC(X)|})$ tests for each X in $TPC(T)$. Two observations to notice are: (i) the complexity order of the first algorithm depends linearly on the size of the problem $|\boldsymbol{V}|$, exponentially on $|PC(T)|$, which is a structural property of the problem, and how close $TPC(T)$ is to $PC(T)$ and (ii) the second algorithm requires multiple times the time of the first algorithm for minimal returns in quality of learning, that is, just to take care of the scenario in Figure 3 and remove the variables $EPC(T) \setminus PC(T)$ (i.e., X in Figure 3).

Since an Oracle is not available the complexity of both algorithms strongly depends on how close approximation of the $PC(T)$ is and how efficiently this approximation is found. The simplest strategy for example is to set $TPC(T) = \boldsymbol{V}$, essentially getting the local version of the algorithm SGS described above. In general any heuristic method that returns a superset of $PC(T)$ is admissible, that is, it could lead to sound algorithms.

Also notice that in the first algorithm the identification of the members of the $TPC(T)$ (step 1) and the removal of variables from it (step 3) can be interleaved. $TPC(T)$ can grow gradually by one, many variables, or all members of it at a time before it satisfies the requirement that is a superset of $PC(T)$. The requirement for the algorithm to be

sound is that, in the end, all tests $I(X, T \mid \boldsymbol{Z})$ for all subsets $\boldsymbol{Z}$ of $PC(T) \setminus \{X\}$ have been performed.

GLL-PC: High-level pseudocode and main components of Generalized Local Learning - Parents and Children. Returns PC(*T*)

1. $\boldsymbol{U} \leftarrow$ GLL-PC-nonsym(T) // first approximate *PC(T)* without symmetry check
2. For all $X \in \boldsymbol{U}$
3. If $T \notin$ GLL-PC-nonsym(X) then $\boldsymbol{U} \leftarrow \boldsymbol{U} \setminus \{X\}$ // check for symmetry
4. Return $\boldsymbol{U}$ // true set of parents and children

GLL-PC-nonsym(*T*) // returns a set which is a subset of $EPC(T)$ and a superset of $PC(T)$

1. Initialization
   a. Initialize a set of candidates for the true $PC(T)$ set: $TPC(T) \leftarrow S$, s.t. $S \subseteq \boldsymbol{V} \setminus \{T\}$
   b. Initialize a priority queue of variables to be examined for inclusion in $TPC(T)$: OPEN $\leftarrow \boldsymbol{V} \setminus \{T \cup TPC(T)\}$
2. Apply inclusion heuristic function
   a. Prioritize variables in OPEN for inclusion in *TPC(T)*;
   b. Throw away non-eligible variables from OPEN;
   c. Insert in *TPC(T)* the highest-priority variable(s) in OPEN and remove them from OPEN
3. Apply elimination strategy to remove variables from *TPC(T)*
4. Apply interleaving strategy by repeating steps #2 and #3 until a termination criterion is met
5. Return *TPC(T)*

Figure 4: High-level outline and main components (underlined) of GLL-PC algorithm.

Given the above, the components of Generalized Local Learning GLL-PC, that is, an algorithm for $PC(T)$ identification based on the above principles are the following: an *inclusion heuristic function* to prioritize variables for consideration as members of $TPC(T)$ and include them in $TPC(T)$ according to established priority. The second component of the framework is an *elimination strategy*, which eliminates variables from the $TPC(T)$ set. An *interleaving strategy* is the third component and it iterates between inclusion and elimination until a stopping criterion is satisfied. Finally the fourth component is the check that the *symmetry requirement* mentioned above is satisfied. See Figure 4 for details. The main algorithm calls an internally defined subroutine that induces parents and children of T without symmetry correction (i.e., returns a set which is a subset of $EPC(T)$ and a superset of $PC(T)$). Note that in all references to $TPC(T)$ hereafter, due to generality of the stated algorithms and the process of convergence of $TPC(T)$ to $PC(T)$, $TPC(T)$ stands for just an approximation to $PC(T)$.

Also notice that the term "priority queue" in the schema of Figure 4 indicates an abstract data structure that satisfies the requirement that its elements are ranked by some priority function so that the highest-priority element is extracted first. $TPC(T)$ in step 1a of the GLL-PC-nonsym subroutine will typically be instantiated with the empty set when no prior knowledge about membership in $PC(T)$ exists. When the user does have prior knowledge indicating that X is a member of $PC(T)$, $TPC(T)$ can be instantiated to contain X. This prior knowledge may come from domain knowledge, experiments, or may be the result of running GLL-PC on variable X and finding that T is in $PC(X)$ when conducting local-to-global learning (Aliferis et al., 2010; Tsamardinos et al., 2006).

Steps #2, 3, 4 in GLL-PC-nonsym can be instantiated in various ways. Obeying a set of specific rules generates what we call "admissible" instantiations. These admissibility rules are given in Figure 5.

Theorem 2 *When the following sufficient conditions hold:*

a. There is a causal Bayesian network faithful to the data distribution P;

b. The determination of variable independence from the sample data D is correct;

c. Causal sufficiency in $\boldsymbol{V}$

any algorithmic instantiation of GLL-PC in compliance with the admissibility rules #1 − #3 *above will return the direct causes and direct effects of T.*

The proof is provided in the Appendix.

We note that the algorithm schema does not address various optimizations and does not address the issue of statistical decisions in finite sample. These will be discussed later. We also note that initialization of $TPC(T)$ in step 1a of the GLL-PC-nonsym function is arbitrary because correctness (unlike efficiency) of the algorithm is not affected by the initial contents of $TPC(T)$.

GLL-PC: Admissibility rules

1. The inclusion heuristic function should respect the following requirement:

   // *Admissibility rule #1*
   All variables $X \in PC(T)$ are eligible for inclusion in the candidate set $TPC(T)$ and each one is assigned a non-zero value by the ranking function. Variables with zero values are discarded and never considered again.

   Note that variables may be re-ranked after each update of the candidate set, or the original ranking may be used throughout the algorithm's operation.

2. The elimination strategy should satisfy the following requirement:

   // *Admissibility rule #2*
   All and only variables that become independent of the target variable T given any subset of the candidate set $TPC(T)$ are discarded and never considered again (whether they are inside or outside $TPC(T)$).

3. The interleaving strategy iterates inclusion and elimination any number of times provided that iterating stops when the following criterion is satisfied:

   //Admissibility rule #3
   At termination no variable outside the set $TPC(T)$ is eligible for inclusion and no variable in the candidate set can be removed at termination.

Figure 5: GLL-PC admissibility rules.

We next instantiate the GLL-PC schema to derive two pre-existing algorithms, interleaved HITON-PC with symmetry correction and MMPC with symmetry correction (Tsamardinos et al., 2006; Aliferis et al., 2003a; Tsamardinos et al., 2003b). Figure 6 depicts the instantiations needed to obtain interleaved HITON-PC.

The interleaved HITON-PC with symmetry correction algorithm starts with an empty set of candidates, then ranks variables for priority for inclusion in the candidate set by univariate association. It discards variables with zero univariate association. It then accepts each variable into $TPC(T)$. If any variable inside the candidate set becomes independent of the response variable T given some subset of the candidate set, then the algorithm removes that variable from the candidate set and never considers it again. In other words, the algorithm attempts to eliminate weakly relevant features from the $TPC(T)$ every time the $TPC(T)$ receives a new member. Iterations of insertion and elimination stop when there are no more variables to examine for inclusion. Once iterating has stopped, the candidate set is filtered using symmetry criterion. Finally, the

Interleaved HITON-PC with symmetry correction
Derived from GLL-PC with following instantiation specifics:

Initialization
TPC(T) ← ∅

Inclusion heuristic function
a. Sort in descending order the variables X in OPEN according to their pairwise association with T, i.e., *Assoc(X, T|∅)*.
b. Remove from OPEN variables with zero association with T, *i.e.*, when I*(X, T|∅)*
c. Insert at end of *TPC(T)* the first variable in OPEN and remove it from OPEN

Elimination strategy
For each $X \in TPC(T)$
If $\exists \boldsymbol{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. I$(X, T|\boldsymbol{Z})$ remove X from *TPC(T)*

Interleaving strategy
Repeat
steps #2 and #3 of GLL-PC-nonsym
Until OPEN=∅

Figure 6: Interleaved HITON-PC with symmetry correction as an instance of GLL-PC.

candidate set is output. Because the admissibility criteria are obeyed, the algorithm is guaranteed to be correct when the assumptions of Theorem 2 hold.

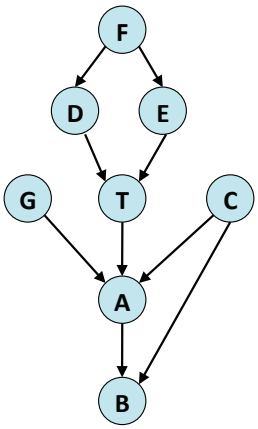


Figure 7: Bayesian network used to trace the algorithms.

Below we prove that that admissibility rules are obeyed in interleaved HITON-PC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero univariate association with T in faithful distributions.

2. Rule #2 (elimination) is directly implemented so it holds.

3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously discarded from OPEN or $TPC(T)$. Neither case can happen because of

Table 1: Trace of GLL-PC-nonsym(T) during execution of interleaved HITON-PC algorithm.

Step of GLL-PC-nonsym	*Comments*	OPEN	*TPC(T)*
1	Initialize *TPC(T)* and OPEN	*{A, B, C, D, E, F, G}*	∅
2a (I)	Prioritize variables in OPEN for inclusion in *TPC(T)*	*{F, D, E, A, B, G, C}*	∅
2b (I)	Throw away non-eligible members of OPEN (*G* and *C*)	*{F, D, E, A, B}*	∅
2c (I)	Insert in *TPC(T)* the highest-priority variable in OPEN (*F*) and remove it from OPEN	*{D, E, A, B}*	*{F}*
3 (I)	Apply elimination strategy to *TPC(T)*: no effect	*{D, E, A, B}*	*{F}*
2 (II)	Insert the highest-priority variable (*D*) in *TPC(T)* and remove it from OPEN	*{E, A, B}*	*{F, D}*
3 (II)	Apply elimination strategy to *TPC(T)*: no effect	*{E, A, B}*	*{F, D}*
2 (III)	Insert the highest-priority variable (*E*) in *TPC(T)* and remove it from OPEN	*{A, B}*	*{F, D, E}*
3 (III)	Apply elimination strategy to *TPC(T)*: remove *F* since I(*T, F*|*{D,E}*)	*{A, B}*	*{D, E}*
2 (IV)	Insert the highest-priority variable (*A*) in *TPC(T)* and remove it from OPEN	*{B}*	*{D, E, A}*
3 (IV)	Apply elimination strategy to *TPC(T)*: no effect	*{B}*	*{D, E, A}*
2 (V)	Insert the highest-priority variable (*B*) in *TPC(T)* and remove it from OPEN	∅	*{D, E, A, B}*
3 (V)	Apply elimination strategy to *TPC(T)*: no effect	∅	*{D, E, A, B}*
4	Stop interleaving since OPEN = ∅	∅	*{D, E, A, B}*

admissibility rules #1, #2 respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

A trace of the algorithm is provided below for data coming out of the example BN of the Figure 7. We assume that the network is faithful and so the conditional dependencies and independencies can be read off the graph directly using the d-separation criterion. Consider that we want to find parents and children of the target variable T using interleaved HITON-PC with symmetry. Table 1 gives a complete trace of step 1 of the instantiated GLL-PC algorithm, that is, execution of GLL-PC-nonsym subroutine for variable T. The Roman numbers in the table refer to iterations of steps 2 and 3 in GLL-PC-nonsym.

Thus we have $TPC(T) = \{D, E, A, B\}$ by the end of GLL-PC-nonsym subroutine, so $\boldsymbol{U} = \{D, E, A, B\}$ in step 1 of GLL-PC. Next, in steps 2 and 3 we first run GLL-PC-nonsym for all $X \in \boldsymbol{U}$:

- GLL-PC-nonsym$(D) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(E) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(A) \rightarrow \{T, G, C, B\}$
- GLL-PC-nonsym$(B) \rightarrow \{A, C\}$

and then check symmetry requirement. Since $T \notin$ GLL-PC-nonsym(B), the variable B is removed from $\boldsymbol{U}$. Finally, the GLL-PC algorithm returns $\boldsymbol{U} = \{D, E, A\}$ in step 4.

Figure 8 shows how algorithm MMPC is obtained from GLL-PC. MMPC is also guaranteed to be sound when the conditions of Theorem 2 hold. Interleaving consists of iterations of just the inclusion heuristic function until OPEN is empty. The heuristic inserts into $TPC(T)$ the next variable F that maximizes the minimum association of variables in OPEN with T given all subsets of $TPC(T)$. In the algorithm, this minimum association of X with T conditioned over all subsets of $\boldsymbol{Z}$ is denoted by $\text{Min}_{\boldsymbol{Z}}\text{Assoc}(X, T \mid \boldsymbol{Z})$. The intuition is that we accept next the variable that despite our best efforts to be made conditionally independent of T (i.e., conditioned on all subsets of our current estimate $TPC(T)$) is still highly associated with T. The two main differences of the MMPC algorithm from interleaved HITON-PC are the more complicated inclusion heuristic function and the absence of interleaving of the inclusion-exclusion phases before all variables have been processed by the inclusion heuristic function. A set of optimizations and caching operations render the algorithm efficient; for complete details see Tsamardinos et al. (2006, 2003b).

Below we prove that admissibility rules are obeyed in MMPC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero conditional association with T in faithful distributions.

2. Rule #2 (elimination) is directly implemented so it holds.

3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously discarded from OPEN or $TPC(T)$. Neither case can happen because of admissibility rules #1, #2 respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

We now introduce a new algorithm, semi-interleaved HITON-PC with symmetry correction, see Figure 9. Semi-interleaved HITON-PC operates like interleaved HITON-PC with one major difference: it does not perform a full variable elimination in $TPC(T)$ with each $TPC(T)$ expansion. On the contrary, once a new variable is selected for inclusion, it attempts to eliminate it and if successful it discards it without further attempted eliminations. If it is not eliminated, it is added to the end of the $TPC(T)$ and new candidates for inclusion are sought. Because the admissibility criteria are obeyed the algorithm is guaranteed to be correct under the assumptions of Theorem 2.

Below we prove that admissibility rules are obeyed in semi-interleaved HITON-PC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero univariate association with T in faithful distributions.

2. Rule #2 (elimination) is directly implemented so it holds.

3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously discarded from OPEN or $TPC(T)$. Neither case can happen because of admissibility rules #1, #2 respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

A trace of the algorithm is provided below for data coming out of the example faithful BN of the Figure 7. Consider that we want to find parents and children of the

MMPC with symmetry correction
Derived from GLL-PC with following instantiation specifics:

Initialization
TPC(T) ← ∅

Inclusion heuristic function
a. Sort in descending order the variables X in OPEN according to $\text{Min}_{\boldsymbol{Z}}\text{Assoc}(X, T|\boldsymbol{Z})$ for $\boldsymbol{Z} \subseteq TPC(T)\backslash\{X\}$
b. Remove from OPEN variables X with zero association with T, given some $\boldsymbol{Z} \subseteq TPC(T)\backslash\{X\}$
c. Insert at end of *TPC(T)* the first variable in OPEN and remove it from OPEN

Elimination strategy
If OPEN=∅
For each $X \in TPC(T)$
If $\exists\, \boldsymbol{Z} \subseteq TPC(T)\backslash\{X\}$, s.t. $I(X, T|\boldsymbol{Z})$ remove X from *TPC(T)*

Interleaving strategy
Repeat
steps #2 and #3 of GLL-PC-nonsym
Until OPEN=∅

Figure 8: MMPC with symmetry correction as an instance of GLL-PC.

Semi-Interleaved HITON-PC with symmetry correction
Derived from GLL-PC with following instantiation specifics:

Initialization
TPC(T) ← ∅

Inclusion heuristic function
a. Sort in descending order the variables X in OPEN according to their pairwise association with T, i.e., $Assoc(X, T|\emptyset)$.
b. Remove from OPEN variables with zero association with T, *i.e.,* when $I(X, T|\emptyset)$
c. Insert at end of *TPC(T)* the first variable in OPEN and remove it from OPEN

Elimination strategy
If OPEN=∅
For each $X \in TPC(T)$
If $\exists\, \boldsymbol{Z} \subseteq TPC(T)\backslash\{X\}$, s.t. $I(X, T|\boldsymbol{Z})$ remove X from *TPC(T)*
Else
X ← last variable added to *TPC(T)* // in step 2 of GLL-PC-nonsym
If $\exists\, \boldsymbol{Z} \subseteq TPC(T)\backslash\{X\}$, s.t. $I(X, T|\boldsymbol{Z})$ remove X from *TPC(T)*

Interleaving strategy
Repeat
steps #2 and #3 of GLL-PC-nonsym
Until OPEN=∅

Figure 9: Semi-interleaved HITON-PC with symmetry correction as an instance of GLL-PC.

target variable T using semi-interleaved HITON-PC with symmetry. Table 2 gives a complete trace of step 1 of the instantiated GLL-PC algorithm, that is, execution of GLL-PC-nonsym subroutine for variable T. The Roman numbers in the table refer to iterations of steps 2 and 3 in GLL-PC-nonsym.

Thus we have $TPC(T) = \{D, E, A, B\}$ by the end of GLL-PC-nonsym subroutine, so $\boldsymbol{U} = \{D, E, A, B\}$ in step 1 of GLL-PC. Next, in steps 2 and 3 we first run GLL-PC-nonsym for all $X \in \boldsymbol{U}$:

Table 2: Trace of GLL-PC-nonsym(T) during execution of semi-interleaved HITON-PC algorithm.

Step of GLL-PC-nonsym	*Comments*	OPEN	*TPC(T)*
1	Initialize *TPC(T)* and OPEN	*{A, B, C, D, E, F, G}*	∅
2a (I)	Prioritize variables in OPEN for inclusion in *TPC(T)*	*{F, D, E, A, B, G, C}*	∅
2b (I)	Throw away non-eligible members of OPEN (*G* and *C*)	*{F, D, E, A, B}*	∅
2c (I)	Insert in *TPC(T)* the highest-priority variable in OPEN (*F*) and remove it from OPEN	*{D, E, A, B}*	*{F}*
3 (I)	Apply elimination strategy to *TPC(T)*: no effect	*{D, E, A, B}*	*{F}*
2 (II)	Insert the highest-priority variable (*D*) in *TPC(T)* and remove it from OPEN	*{E, A, B}*	*{F, D}*
3 (II)	Apply elimination strategy to *TPC(T)*: no effect	*{E, A, B}*	*{F, D}*
2 (III)	Insert the highest-priority variable (*E*) in *TPC(T)* and remove it from OPEN	*{A, B}*	*{F, D, E}*
3 (III)	Apply elimination strategy to *TPC(T)*: No effect	*{A, B}*	*{F, D, E}*
2 (IV)	Insert the highest-priority variable (*A*) in *TPC(T)* and remove it from OPEN	*{B}*	*{F, D, E, A}*
3 (IV)	Apply elimination strategy to *TPC(T)*: no effect	*{B}*	*{F, D, E, A}*
2 (V)	Insert the highest-priority variable (*B*) in *TPC(T)* and remove it from OPEN	∅	*{F, D, E, A, B}*
3 (V)	Apply elimination strategy to *TPC(T)*: remove *F* since I(*T, F*|*{D,E}*)	∅	*{D, E, A, B}*
4	Stop interleaving since OPEN = ∅	∅	*{D, E, A, B}*

- GLL-PC-nonsym$(D) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(E) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(A) \rightarrow \{T, G, C, B\}$
- GLL-PC-nonsym$(B) \rightarrow \{A, C\}$

and then check symmetry requirement. Since $T \in$ GLL-PC-nonsym(B), the variable B is removed from $\boldsymbol{U}$. Finally, the GLL-PC algorithm returns $\boldsymbol{U} = \{D, E, A\}$ in step 4.

4.2. Discovery of the $MB(T)$ Set

As mentioned in Section 3 the $MB(T)$ contains all information sufficient for the determination of the conditional distribution of T : $P(T \mid MB(T)) = P(T \mid \boldsymbol{V} \setminus \{T\})$ and further, it coincides with the parents, children and spouses of T in any network faithful to the distribution (if any) under causal sufficiency. The previous subsection described a general family of algorithms to obtain the $PC(T)$ set, and so in order to find the $MB(T)$ one needs in addition to $PC(T)$, to also identify the spouses of T.

First notice that, approximating $MB(T)$ with $PC(T)$ and missing the spouse nodes may in theory discard very informative nodes. For example, suppose that X and T are two uniformly randomly chosen numbers in $[0, 1]$ and that $Y = \min(1, X + T)$. Then, the only faithful network representing the joint distribution is $X \rightarrow Y \leftarrow T$, where X is the spouse of T. In predicting T, the spouse node X may reduce the uncertainty

completely: conditioned on Y, T may become completely determined (when both X and T are less than 0.5). Thus, it theoretically makes sense to develop algorithms that identify the spouses in addition to the $PC(T)$, even though later in Section 5 we empirically determine that within the scope of distributions and problems tried, the $PC(T)$ resulted in feature subsets almost as predictive as the full $MB(T)$. In the companion paper (Aliferis et al., 2010) we also provide possible reasons explaining the good performance of $PC(T)$ versus $MB(T)$ for classification in practical tasks.

The theorem on which the algorithms in this family are based to discover the $MB(T)$ is the following:

Theorem 3 *In a faithful BN $\langle \boldsymbol{V}, G, P \rangle$, if for a triple of nodes X, T, Y in G, $X \in PC(Y)$, $Y \in PC(T)$, and $X \notin PC(T)$, then $X \rightarrow Y \leftarrow T$ is a subgraph of G iff $\neg I(X, T \mid \boldsymbol{Z} \cup \{Y\})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (Spirtes et al., 2000).*

We distinguish two cases: (i) X is a spouse of T but it is also a parent or child, for example, $X \rightarrow T \rightarrow Y$ and also $X \rightarrow Y$. In this case, we cannot use the theorem above to identify Y as a collider and X as a spouse. But at the same time we do not have to: $X \in PC(T)$ and so it will be identified by GLL-PC. (ii) $X \in MB(T) \setminus PC(T)$ in which case we can use the theorem to locally discover the subgraph $X \rightarrow Y \leftarrow T$ and determine that X should be included in $MB(T)$.

We now introduce the GLL-MB in Figure 10. The admissibility requirement is simply to use an admissible GLL-PC instantiation.

GLL-MB: Generalized Local Learning - Markov Blanket

1. *PC(T)* ← GLL-PC*(T)* // obtain *PC*(*T*) by running *GLL-PC* for variable *T*
2. For every variable $Y \in PC(T)$
   PC(*Y*) ← GLL-PC(*Y*) // obtain *PC*(*Y*) for every member *Y* of *PC*(*T*)
3. *TMB(T)* ← *PC(T)* // initialize *TMB(T)* with *PC*(*T*) members
4. $\boldsymbol{S} \leftarrow \{\cup_{Y \in PC(T)} PC(Y)\} \setminus \{PC(T) \cup \{T\}\}$ // these are the potential spouses
5. For every variable $X \in \boldsymbol{S}$
   a. Retrieve a subset $\boldsymbol{Z}$ s.t. I(*X*, *T* | $\boldsymbol{Z}$) // subset was identified and stored in steps 1 and 2
   b. For every variable $Y \in PC(T)$ *s.t.* $X \in PC(Y)$ // *Y* is a potential common child of *T* and *X*
   c. If $\neg$I(*X*, *T*| $\boldsymbol{Z} \cup \{Y\}$) // *X* is a spouse
   d. Insert *X* into *TMB(T)*
6. Optionally: Eliminate from *TMB(T)* predictively redundant members using a backward wrapper approach.
7. Return *TMB*(*T*)

Figure 10: GLL-MB: Generalized Local Learning – Markov Blanket algorithm.

For the identification of $PC(T)$ any method of GLL-PC can be used. Also, in step 5a we know such a $\boldsymbol{Z}$ exist since $X \notin PC(T)$ (by Theorem 1); this $\boldsymbol{Z}$ has been previously determined and is cached during the call to GLL-PC.

Theorem 4 *When the following sufficient conditions hold*

a. There is a causal Bayesian network faithful to the data distribution P;

b. The determination of variable independence from the sample data D is correct;

c. Causal sufficiency in $\boldsymbol{V}$

any algorithmic instantiation of GLL-MB in compliance with the admissibility rule will return $MB(T)$ (with no need for step 6).

The proof is provided in the Appendix.

A new Markov blanket algorithm, semi-interleaved HITON-MB, can be obtained by instantiating GLL-MB (Figure 10) with the semi-interleaved HITON-PC algorithm with symmetry correction for GLL-PC.

Semi-interleaved HITON-MB is guaranteed to be correct under the assumptions of Theorem 4, hence the only proof of correctness needed is the proof of correctness for semi-interleaved HITON-PC with symmetry (which was provided earlier).

A trace of the semi-interleaved HITON-MB algorithm for data coming out of the example faithful BN of the Figure 7 follows below. Please refer to Figure 10 for step numbers. Consider that we want to find Markov blanket of T. In step 1, we find $PC(T) = \{D, E, A\}$. Then in step 2 we find $PC(X)$ for all $X \in PC(T)$:

- $PC(D) = \{T, F\}$,
- $PC(E) = \{T, F\}$,
- $PC(A) = \{T, G, C, B\}$,

In step 3 we initialize $TMB(T) \leftarrow \{D, E, A\}$. The set $\boldsymbol{S}$ in step 4 contains the following variables: $\{F, G, C, B\}$. In step 5 we loop over all members of $\boldsymbol{S}$ to find spouses of T. Let us consider each variable separately:

- Loop for $X = F$: In step 5a we retrieve a subset $\boldsymbol{Z} = \{D, E\}$ that renders $X = F$ independent of T. In step 5b we loop over all potential common children of F and T, that is, $Y = D$ and $Y = E$. When we consider $Y = D$, we find that $X = F$ is independent of T given $\boldsymbol{Z} \cup \{Y\} = \{D, E\}$ and thus do not include F in $TMB(T)$ in step 5d. When we consider $Y = E$, we also do not include F in $TMB(T)$ in step 5d.
- Loop for $X = G$: In step 5a we retrieve a subset $\boldsymbol{Z} = \emptyset$ that renders $X = G$ independent of T. In step 5b we loop over all potential common children of G and T, that is, variable $Y = A$. We find that $X = G$ is dependent on T given $\boldsymbol{Z} \cup \{Y\} = \{A\}$ and thus include G in $TMB(T)$ in step 5d.
- Loop for $X = C$: In step 5a we retrieve a subset $\boldsymbol{Z} = \emptyset$ that renders $X = C$ independent of T. In step 5b we loop over all potential common children of C and T, that is, variable $Y = A$. We find that $X = C$ is dependent on T given $\boldsymbol{Z} \cup \{Y\} = \{A\}$ and thus include C in $TMB(T)$ in step 5d.
- Loop for $X = B$: In step 5a we retrieve a subset $\boldsymbol{Z} = \{A, C\}$ that renders $X = B$ independent of T. In step 5b we loop over all potential common children of B and T, that is, variable $Y = A$. We find that $X = B$ is independent of T given $\boldsymbol{Z} \cup \{Y\} = \{A, C\}$ and thus do not include G in $TMB(T)$ in step 5d.

By the end of step 5, we have $TMB(T) = \{D, E, A, G, C\}$. Notice that it is the true $MB(T)$. In step 6 we perform wrapping to remove members of $TMB(T)$ that are redundant for classification. Let us assume that we used a backward wrapping procedure that led to removal of variable G, for example because omitting this variable does not increase classification loss. Thus, we have $TMB(T) = \{D, E, A, C\}$ in step 7 when the algorithm terminates.

The above algorithm specifications and proofs demonstrate that it is relatively straightforward to derive correct algorithms and prove their correctness using the GLL framework. It is also straightforward to derive relaxed versions (for example non-symmetry corrected versions of interleaved and semi-interleaved HITON and MMPC) which trade-off correctness for improved tractability.

4.3. Computational Complexity

The complexity of all algorithms presented depends on the time for the tests of independence and measures of associations. For the G^2 test of independence for discrete variables, for example, we use in reported experiments an implementation linear to the sample size and exponential to the number of variables in the conditional set. However, because the latter number is small in practice, tests are relatively efficient. Faster implementations exist that only take time $n \log(n)$ to the number n of training instances, independent of the size of the conditioning set. Also, advanced data structures (Moore and Wong, 2003) can be employed to improve the time complexity (see Tsamardinos et al. 2006 for details on the implementation of the tests). In reported experiments we also implement the measure of association $\text{Assoc}(X, T \mid \boldsymbol{Z})$ to be the negative *p-value* returned by the test $I(X, T \mid \boldsymbol{Z})$ and so it takes exactly the same time to compute as a test of independence. In the following discussion, we consider the complexity of the algorithms in terms of the number of tests and measures of association they perform.

The number of tests of the GLL-PC algorithm in Figure 4 depends on several factors. These are the inclusion heuristic efficiency in approximating the $PC(T)$, the time required by the inclusion heuristic, and the size of the $PC(T)$ which is a structural property of the problem to solve. Interleaved-HITON-PC (algorithm in Figure 6) for example, will sort the variables using $|\boldsymbol{V}|$ measures of associations. Subsequently, it will perform a test $I(X, T \mid \boldsymbol{Z})$ for all subsets of the largest $TPC(T)$ in any iteration of interleaving of the inclusion-exclusion steps. With appropriate caching a test will never have to be repeated. Thus, assuming the largest size of the $TPC(T)$ is in the order of the $PC(T)$, the complexity of the GLL-PC-nonsym subroutine is $O(|\boldsymbol{V}|2^{|PC(T)|})$. In step 3, it will execute the GLL-PC-nonsym subroutine again for all $X \in TPC(T)$. Assuming each neighborhood of X is about the same as the $PC(T)$, when checking the symmetry condition, the algorithm will perform another $O(|\boldsymbol{V}||PC(T)|2^{|PC(T)|})$tests.

To identify $MB(T)$ by the GLL-MB algorithm in Figure 10 we first need to initialize subset $\boldsymbol{S}$. Assuming all neighborhoods are about the same size (i.e., equal to $|PC(T)|$), the total complexity to find the set $\boldsymbol{S}$ is $O(|\boldsymbol{V}||PC(T)|^2 2^{|PC(T)|})$ since we call GLL-PC for each member of the $PC(T)$. In fact, several optimizations can reduce this order to $O(|\boldsymbol{V}||PC(T)|2^{|PC(T)|})$ but we will not elaborate further in this paper. In step 5, in the worst case we perform a single test for each node in $\boldsymbol{S}$ and each node in $PC(T)$ for a total of at most $O(|PC(T)|^2)$ tests (the subset $\boldsymbol{Z}$ in step 5a is cached and retrieved). So the order of the algorithm is $O(|\boldsymbol{V}||PC(T)|^2 2^{|PC(T)|})$ tests given the structural assumptions above.

All other algorithmic instantiations of the template in this section have similar complexity.

At this point it is worth noting a number of polynomial approximation algorithms in the literature that increase efficiency without sacrificing quality to a large degree. The identification of a subset $\boldsymbol{Z}$ in step 3 of the GLL-PC-nonsym subroutine as described in algorithm instantiations of GLL-PC is a step exponential to the size of the $TPC(T)$; however, one could attempt to discover it in a greedy fashion, for example by starting

with the empty set and adding to $\boldsymbol{Z}$ the variable decreasing the association with T the most. These ideas started with the TPDA algorithm (Cheng et al., 2002a) and were further explored in Brown et al. (2005). Similar improvements can be applicable to inclusion strategy.

For the above analysis we assumed that all tests $I(X, T \mid \boldsymbol{Z})$ can or should be performed and return the correct results. However, in the next sub-section we discuss how the statistical decisions of independence or dependence are made; these decisions severely affect the complexity of the algorithms as well.

4.4. Dealing with Statistical Decisions

The quality of the algorithms in practice highly depends on their ability to statistically determine whether $I(X, T \mid \boldsymbol{Z})$ or $\neg I(X, T \mid \boldsymbol{Z})$ (equivalently whether $\text{Assoc}(X, T \mid \boldsymbol{Z})$ is zero or non-zero) for a pair of variables X and T and a set of variables Z. The test $I(X, T \mid \boldsymbol{Z})$ is implemented as a statistical hypothesis test with null hypothesis H_0: X and T are independent given $\boldsymbol{Z}$. A *p-value* corresponding to this test statistic's distribution expresses the probability of seeing the same or more extreme (i.e., indicative of dependence) test statistic values when sampling from distributions where H_0 is true. If the *p-value* is lower than a given threshold (i.e., significance level "alpha") α, then we consider the independence hypothesis to be improbable and reject it. Thus, for a sufficiently low *p-value* we accept *dependence*. If however, the *p-value* is not low enough to provide confidence in rejecting H_0 then there are two possibilities:

a) H_0 actually holds, that is, the variables are indeed conditionally independent.

b) H_0 does not hold, the variables are conditionally dependent but we cannot confidently reject H_0.

The reasons for b) are that either the dependence is weak relatively to the available sample to be detected (in order words, we have low probability to reject the null hypothesis H_0 when it does not hold, that is, low statistical power), or we are using the wrong statistical test for this type of dependency. In essence, we would like to distinguish between the following cases:

a) $I(X, T \mid \boldsymbol{Z})$ holds with high-probability

b) $\neg I(X, T \mid \boldsymbol{Z})$ holds with high-probability

c) Undetermined case given the available sample

To deal with case c) in our implementations we take the following approach, introduced by Spirtes et al. (2000): we consider that we are facing case c) if there is no sufficient power according to a *reliability criterion*. In our implementations this criterion depends on parameter *h-ps*. The criterion dictates that if and only if we have at least *h-ps* sample instances per number of cells (i.e., number of parameters to be estimated) in the contingency tables for the discrete statistical tests then the test is reliable.

Once a test is deemed unreliable an algorithm needs to decide how to handle the corresponding statistical decision. For example, the PC algorithm for global causal discovery (Spirtes et al., 2000) considers that given no other evidence, all variables are dependent with each other. That is, a pair of variables is always connected by an edge in the graph unless a subset $\boldsymbol{Z}$ is discovered that renders them conditionally independent.

The implementations of GLL instantiations in the present paper do not perform an unreliable test either. However, ignoring unreliable tests with 0-order conditioning test (i.e., univariate tests) is equivalent to assuming $I(X, T \mid \boldsymbol{Z})$ whereas ignoring unreliable tests with higher-order conditioning test (i.e., conditioning sets with 1 or more conditioning variables) is equivalent to assuming $\neg I(X, T \mid \boldsymbol{Z})$ as far as this unreliable test is concerned (because the final judgment on independence, is deferred to reliable, typically lower-order tests). Thus, given no evidence of dependence, we assume the unreliable tests to return $I(X, T \mid \boldsymbol{Z})$. The different treatment of the PC implementation leads to problems as discussed in Tsamardinos et al. (2006) pointing to the importance of this implementation aspect of the algorithms.

Another practical implementation issue arises when prior knowledge, experiments, or domain substantive knowledge ensures that a variable X is in $PC(T)$ or that X is not in $PC(T)$. In such cases the algorithm can be modified to "lock" X inside or outside $TPC(T)$ respectively in order to avoid the possibility that errors in statistical decisions will counter previously validated knowledge and possibly propagate more statistical decision errors.

In addition to *h-ps*, a second restriction on the conditioning set size is provided by parameter *max-k*. This parameter places an absolute limit on the number of elements in a conditioning set size, *without reference to available sample size*. As such *max-k* participates in the reliability judgment but also restricts the computational complexity of the algorithms by trading off computational complexity for fit to data.

Specifically first consider that more variables than the actual $PC(T)$ could be output by the algorithm. A variable X that becomes independent of T only when we condition on $\boldsymbol{Z}$, with $|\boldsymbol{Z}| > max\text{-}k$ could enter the $TPC(T)$ and will not be removed afterwards. For example, if $max\text{-}k = 1$, then variable F in Figure 7 cannot be d-separated from T given any $\boldsymbol{Z}$ with $|\boldsymbol{Z}| \leq 1$. Thus, the reliability criterion may increase the number of tests performed, since these depend on the size of the $TPC(T)$. On the other hand, the criterion forces certain tests not to be performed, specifically those whose conditioning set $\boldsymbol{Z}$ size is larger than *max-k*. Thus, since only $\binom{TPC(T)}{max\text{-}k}$ subsets are tested out of all possible $2^{|TPC(T)|}$ ones, the complexity of the algorithm GLL-PC-nonsym now becomes $O(|\boldsymbol{V}||TPC(T)|^{max\text{-}k})$, that is, polynomial of order *max-k*.

The parameters *h-ps* and *max-k* are user-specified or, alternatively, optimized automatically by cross-validation, or optimized for a whole domain. The role and importance of these two parameters, especially with respect to quality of statistical decisions, is explored in detail in the companion paper (Aliferis et al., 2010). Finally, because the quality of statistical decisions is not addressed in the proofs of correctness provided earlier, it was implicitly assumed that whenever sufficient sample size is provided to the algorithms statistical decisions are reliable.

A recent treatment that specifically addresses the role of statistical decisions in finite sample is presented in Tsamardinos and Brown (2008a). In this work, a bound of the p-value of the existence of an edge is provided; the bound can be used to control the False Discovery Rate of the identification of the $PC(T)$ or all the edges in a network.

5. Comparative Evaluation of Local Causal and Non-Causal Feature Selection Algorithms in Terms of Feature Selection Parsimony and Classification Accuracy

In the present section we examine the ability of GLL algorithms to discover compact sets of features with as high classification performance as possible for each data set and compare them with other local causal structure discovery methods as well as non-causal feature selection methods.

In order to avoid bias in error estimation we apply nested N-fold cross-validation. The inner loop is used to try different parameters for the feature selection and classifier methods while the outer loop tests the best configuration on an independent test set. Details are given in Statnikov et al. (2005b), Dudoit and van der Laan (2003) and Scheffer (1999).

All experiments discussed in this section and elsewhere in this paper were conducted on ACCRE (Advanced Computing Center for Research and Education) High Performance Computing system at Vanderbilt University. The ACCRE system consists of 924 x86 processors (the majority of which 2 GHz) and 668 PowerPC processors (2.2 GHz) running 32 and 64-bit Linux OS. The overall computational capacity of the cluster is approximately 6 TFLOPS. For preliminary and exploratory experiments we used a smaller cluster of eight 3.2 GHz x86 processors.

The evaluated algorithms are listed in the Appendix Tables 5–7. They were chosen on the basis of prior independently published results showing their state-of-the-art performance and applicability to the range of domains represented in the evaluation data sets. We compare several versions of GLL, including parents and children (PC) and Markov blanket (MB) inducers. Whenever we refer to HITON-PC algorithm in this paper, we mean semi-interleaved HITON-PC without symmetry correction, unless mentioned otherwise. Also, other GLL algorithms evaluated do not have symmetry correction unless mentioned otherwise. Finally, unless otherwise noted, GLL-MB does not implement a wrapping step.

Tables 8–9 in the Appendix present the evaluation data sets. The data sets were chosen on the basis of being representative of a wide range of problem domains (biology, medicine, economics, ecology, digit recognition, text categorization, and computational biology) in which feature selection is essential. These data sets are challenging since they have a large number of features with small-to-large sample sizes. Several data sets used in prior feature selection and classification challenges were included. All data sets have a single binary target variable.

To perform imputation in data sets with missing values, we applied a non-parametric nearest neighbor method (Batista and Monard, 2003). Specifically, this method imputes each missing value of a variable with the present value of the same variable in the most similar instance according to Euclidian distance metric. Discretization in non-sparse continuous data sets was performed by a univariate method (Liu et al., 2002) implemented in *Causal Explorer* (Aliferis et al., 2003b). For a given continuous variable, the method considers many binary and ternary discretization thresholds (by means of a sliding window) and chooses the one that maximizes statistical association with the target variable. In sparse continuous data sets, discretization was performed by assigning value 1 to all non-zero values. All variables in each data set were also normalized to be in [0, 1] range to facilitate classification by SVM and KNN. All computations of statistics for the preprocessing steps were performed based on training data only to ensure unbiased classification error estimation. Statistical comparison between algorithms was done

using two-sided permutation test (with 10,000 permutations) at 5% alpha level (Good, 2000). The null hypothesis of this test is that algorithms perform the same.

Both polynomial SVMs and KNN were used for building classifiers from each selected feature set. In complementary experiments, the *native* classifier for each one of several feature selection methods (LARS-EN, L0, and RFVS) was used and its performance was compared against classifiers induced by SVMs and KNN. For SVMs, the misclassification cost C and kernel degree d were optimized over values [1, 10, 100] and [1, 2, 3, 4], respectively. For KNN, the number of nearest neighbors k was optimized over values [1,...,min(1000, number of instances in the training set)]. All optimization was conducted in nested cross-validation using training data only, while the testing data was used only once to obtain an error estimate for the final classifier. We used the libSVM implementation of SVM classifiers (Fan et al., 2005) and our own implementation of KNN.

We note that use of SVMs and KNN does not imply that GLL methods are designed to be filters for these two algorithms only, or that the algorithm comparison results narrowly apply to these two classifiers. Rather as explained in Section 2.2, GLL algorithms provide performance guarantees as long as the classifier used has universal approximator properties. SVMs and KNN are two exemplars of practical and scalable such methods in wide use. We also emphasize that selecting features with a wrapper or embedded feature selection method that *is not* SVM or KNN specific is not affected by the inductive bias mismatch because such mismatch is affecting performance only when the classifier used is "handicapped" relative to the native classifier (Tsamardinos and Aliferis, 2003; Kohavi and John, 1997). We provide experimental data substantiating this point in the Appendix Table 10 (and Table S1 in the online supplement) where we compare classification performance of RFVS, LARS-EN, and L0 with features selected by each corresponding method to the classification performance of SVMs and KNN using the same features. It is shown that SVM predictivity matches, whereas KNN predictivity compares favorably, with the classifiers that are native to each feature selector. On the other hand, the choice of SVMs and KNN provides several advantages to the research design of the evaluation: (a) the same classifiers can be used with all data sets removing a possible confounder in the evaluation; (b) they can be used without feature selection (i.e., full variable set) to give a reference point of predictivity under no feature selection (that in practice is as good as empirically optimal predictivity especially when using SVMs); (c) they can be used when sample size is smaller than number of variables; (d) prior evidence suggests that they are suitable classifiers for the domains; (e) they can be executed in tractable time using nested cross-validation as required by our protocol.

In all cases when an algorithm had not terminated within 2 days of single-CPU time per run on a single training set (including optimization of the feature selector parameters) and in order to make the experimental comparison feasible with all methods and data sets in the study, we deemed it to be impractical and terminated it. While the practicality of spending more than two days of single-CPU time on a single training set can be debated, we believe that use of slower algorithms in practice is problematic due to the following reasons: (i) in the context of N-fold cross-validation the total running time is at least N times longer (i.e., $>$20 days single-CPU time); (ii) the analyst does not know whether the algorithm will terminate within a reasonable amount of time, and (iii) when quantification of uncertainty about various parameters (e.g., estimating variance in error estimates via bootstrapping) is needed the analysis becomes prohibitive regardless of analyst flexibility and computational resources. When comparing a pair of

algorithms we consider only the data sets where both algorithms terminate within the allotted time.

We evaluate the algorithms using the following metrics:

1. Number of features selected;
2. Proportion of features selected relative to the original number of features (i.e., prior to feature selection);
3. Classification performance measured as area under ROC curve (AUC) (Fawcett, 2003);
4. Feature selection time in minutes.[2]

Figure 11 compares each evaluated algorithm to semi-interleaved HITON-PC with G^2 test as a reference performance for GLL, in the two-dimensional space defined by proportion of selected features and classification performance by SVM (results for KNN are similar and are available in Table S5 in the online supplement). As can be seen in the figure (and also in Figure S1 of the online supplement), GLL algorithms typically return much more compact sets than other methods. More compact results are provided by versions that induce the PC set rather than the MB for obvious reasons. Out of GLL methods, the most compact sets are returned when the Z-test is applicable (continuous data) compared to G^2 test (discrete or discretized data). As seen in Tables S2-S3 in the online supplement, depending on the parameterization of GLL, compactness varies. However, regardless of configuration, both GLL and other local causal methods (i.e., IAMB, BLCD-MB, FAST-IAMB, K2MB) with the exception of Koller-Sahami are typically more compact than non-causal feature selection methods (i.e., univariate methods with backward wrapping, RFE, RELIEF, Random Forest-based Variable Selection, L0, and LARS-EN). Forward stepwise selection and some configurations of LARS-EN, Random Forest-based Variable Selection, and RFE are often very parsimonious, however their parsimony varies greatly across data sets. Notice that whenever an algorithm variant employed statistical comparison among feature sets (in particular non-causal ones), it improved compactness (Figure S1 and Tables S2-S3 in the online supplement). Table 3 gives statistical comparisons of compactness between one reference GLL algorithm (semi-interleaved HITON-PC with G^2 test and cross-validation-based optimization of the algorithm parameters) and 43 non-GLL algorithms and variants (including no feature selection). In 21 cases the GLL reference method gives statistically significantly more compact sets compared to all other methods, in 16 cases parsimony is not statistically distinguishable, and in 6 cases HITON-PC gives less compact feature sets. These 6 cases correspond strictly to non-GLL causal feature selection algorithms and at the expense of severe predictive suboptimality (0.06 to 0.10 AUC) relative to the reference GLL method (see Tables S4-S5 in the online supplement).

5.1. Compactness Versus Classification Performance

Compactness is only one of the two requirements for solving the feature selection problem. A maximally compact algorithm that does not achieve optimal predictivity

2. In all cases we used the implementations provided by the authors of methods, or state-of-the-art implementations, and thus reported time should be considered representative of what practitioners can expect in real-life with equipment and data similar to the ones used in the present study. However, we note that running times should be interpreted as indicative only since numerous implementation details and possible optimizations as well as computer platform discrepancies can affect results.

Table 3: Statistical comparison via permutation test (Good, 2000) of 43 non-GLL algorithms (including no feature selection) to the reference GLL algorithm (semi-interleaved HITON-PC with G^2 test and cross-validation-based optimization of the algorithm parameters by SVM classifier) in terms of SVM predictivity and parsimony. Each non-GLL algorithm compared to HITON-PC in each row is denoted by "Other". Bolded p-values are statistically significant at 5% alpha.

	Predicitivity		***Reduction***	
Feature selection method	P-value	Nominal winner	P-value	Nominal winner
No feature selection	0.1890	Other	**<0.0001**	HITON-PC
RFE: 4 variants	0.9754	Other	**0.0046**	HITON-PC
	0.8030	Other	**0.0042**	HITON-PC
	0.1312	HITON-PC	0.3634	HITON-PC
	0.1008	HITON-PC	0.6816	Other
UAF-KruskalWallis-SVM: 4 variants	0.2248	Other	**0.0028**	HITON-PC
	0.0098	Other	**0.0004**	HITON-PC
	1.0000	HITON-PC	0.1414	HITON-PC
	0.3232	HITON-PC	0.3998	HITON-PC
UAF-Signal2Noise-SVM: 4 variants	0.0710	Other	**0.0018**	HITON-PC
	0.0752	Other	**0.0030**	HITON-PC
	0.4420	HITON-PC	0.7850	HITON-PC
	0.2820	HITON-PC	0.6604	HITON-PC
UAF-Neal-SVM: 4 variants	0.5046	Other	**<0.0001**	HITON-PC
	0.9782	HITON-PC	**<0.0001**	HITON-PC
	0.6980	HITON-PC	**0.0044**	HITON-PC
	0.3806	HITON-PC	**0.0186**	HITON-PC
Random Forest Variable Selection: 2 variants	0.6064	HITON-PC	0.3252	HITON-PC
	0.5050	HITON-PC	0.1338	Other
LARS-Elastic Net: 2 variants	1.0000	Other	0.1112	HITON-PC
	0.0832	HITON-PC	0.5216	Other
RELIEF: 8 variants	0.2032	Other	**<0.0001**	HITON-PC
	0.9362	Other	**<0.0001**	HITON-PC
	0.4388	Other	**0.0014**	HITON-PC
	0.8432	Other	**0.0010**	HITON-PC
	0.4290	HITON-PC	**0.0108**	HITON-PC
	0.3114	HITON-PC	0.0518	HITON-PC
	0.4424	HITON-PC	0.0706	HITON-PC
	0.2748	HITON-PC	**0.0404**	HITON-PC
L0-norm	**0.0258**	HITON-PC	0.1942	HITON-PC
Forward Stepwise Selection	**0.0028**	HITON-PC	0.2758	Other
Koller-Sahami: 6 variants	0.7506	HITON-PC	**<0.0001**	HITON-PC
	0.6234	HITON-PC	**<0.0001**	HITON-PC
	0.6278	HITON-PC	**<0.0001**	HITON-PC
	<0.0001	HITON-PC	**<0.0001**	Other
	0.1278	HITON-PC	0.3856	HITON-PC
	0.1236	HITON-PC	**<0.0001**	HITON-PC
IAMB: 3 variants	**<0.0001**	HITON-PC	**<0.0001**	Other
	<0.0001	HITON-PC	**<0.0001**	Other
	<0.0001	HITON-PC	0.1202	Other
K2MB	**<0.0001**	HITON-PC	**<0.0001**	Other
BLCD-MB	**<0.0001**	HITON-PC	**<0.0001**	Other
FAST-IAMB	**<0.0001**	HITON-PC	**<0.0001**	Other

does not solve the feature selection problem. Figure 11 examines the trade-off of compactness and SVM predictivity (results for KNN are similar and available in Table S5 in the online supplement). The best possible point for each graph is at the upper left corner. For ease of visualization the results are plotted for each algorithmic family separately. To avoid overfitting and to examine robustness of various methods to parameterization we did not select the best performing configuration, but plotted all of them. Notice that some algorithms did not run on all 13 real data sets (i.e., algorithms

with Fisher's Z-test are applicable only to continuous data, while some algorithms did not terminate within 2 days of single-CPU time per run on a single training set). For such cases, we plotted results only for data sets where the algorithms were applicable and the results for HITON-PC correspond to the same data sets. As can be seen, GLL algorithms that induce PC sets dominate both other causal and non-causal feature selection algorithms. This is also substantiated in Table 3 (and Table S7 in the online supplement that provides results for KNN classifier) that gives statistical comparisons of predictivity between the reference GLL algorithm and all 43 non-GLL algorithms and variants (including no feature selection). In 9 cases the GLL reference method gives statistically significantly more predictive sets compared to all other methods, in 33 cases predictivity is not statistically distinguishable, and in 1 case GLL gives less predictive feature sets (however the magnitude of the GLL suboptimal predictivity is only 0.018 AUC on average, whereas the difference in compactness is more than 33% features selected on average).

The overall performance patterns of combined predictivity and parsimony are highly consistent with Markov blanket induction theory (Section 2.2) which predicts maximum compactness and optimal classification performance when using the MB. Different instantiations of the GLL method give different trade-offs between predictivity and parsimony (details and statistical comparisons to the reference method are provided in online supplement Tables S2-S6 and S8).

In the companion paper (Aliferis et al., 2010), we examine in detail conditions under which PC induction can give optimal classification performance (the empirical illustration is provided in Figure 13). The comparison of HITON-PC with G^2 test and HITON-PC with Z-test reveals that both statistics perform similarly, while the latter (where it is applicable) does not require discretization of continuous data that can simplify data analysis significantly (see Figure 12 and statistical comparisons in Table S9 in the online supplement). In Table S10 of the online supplement we provide statistical comparisons of non-GLL causal feature selection methods in terms of predictivity and parsimony. K2MB, BLCD-MB, IAMB, and FAST-IAMB rather unexpectedly perform statistically indistinguishably in terms of predictivity and parsimony. Since BLCD-MB differs from K2MB by an additional backward elimination step, this implies that this step rarely results in elimination of features in the real data sets tested.

5.2. Analysis of Running Times

Table S6 in the online supplement gives detailed running times for all feature selection experiments. Major observations include that: (i) univariate methods, RELIEF, RFE, LARS-EN are in general the fastest ones, (ii) Koller-Sahami is probably the slowest method since it does not terminate on several data sets within the allotted time limit, (iii) FAST-IAMB is two orders of magnitude faster than IAMB on the average, and (iv) GLL algorithms are practical for very high-dimensional data (e.g., in the Thrombin data set with $>$ 100,000 features GLL-PC requires 10 to 52 minutes single-CPU time depending on fixed-parameter configuration, and less than 3 hours when GLL-PC is automatically optimized by cross-validation).

In conclusion, the GLL reference algorithm dominates most feature selection methods in predictivity and compactness. Some non-GLL causal methods are more parsimonious than the reference GLL method at the expense of severe classification suboptimality. One univariate method exhibits slightly higher predictivity but with severe

disadvantage in parsimony. No feature selection method achieves equal or better compactness with equal or better classification performance than GLL.

6. Comparative Evaluation of Markov Blanket Induction, Local Causal Neighborhood and Other Non-Causal Algorithms for Local Structure Discovery

In the present section we study the ability of GLL algorithms to discover local causal structure (in the form of parent and children sets and Markov blankets) and compare them with other local structure discovery methods as well as non-causal feature selection. While many researchers apply feature selection techniques strictly to improve the cost and effectiveness of classification, in many fields researchers routinely apply feature selection in order to gain insights *about the causal structure of the domain*. A frequently encountered example is in bioinformatics where a plethora of feature selection methods are applied in high-throughput genomic and proteomic data to discover biomarkers suitable for new drug development, personalizing medical treatments, and orienting subsequent experimentation (Zhou et al., 2002; Li et al., 2001; Holmes et al., 2000; Eisen et al., 1998). It is thus necessary to test the appropriateness of various feature selection techniques for causal discovery, not just classification.

In order to compare the performance of the tested techniques for causal discovery, we simulate data from known Bayesian networks and also use resimulation, whereby real data is used to elicit a causal network and then data is simulated from the obtained network (see Table 11 in the Appendix). For each network, we randomly select 10 different targets and generate 5 samples (except for sample size 5,000 where one sample is generated) to reduce variability due to sampling.[3] An independent sample of 5,000 instances is used for evaluation of classification performance.

In order to avoid overfitting of the results to the method used to induce the causal network, an algorithm with different inductive bias is used than the algorithms tested. In our case we use SCA (Friedman et al., 1999b). We note that SCA has greatly different inductive bias from the GLL variants and thus the comparison (provided that the causal generative model is a Bayesian network) is not unduly biased toward them, while still allowing induction of a credible causal graphical model. Specifically, the inductive biases of the two methods can be described as follows: SCA performs global, heuristically constrained, Bayesian search-and-score, greedy TABU iterative search for a Bayesian network that has maximum-a-posteriori probability given the data under uninformative prior on all possible network structures. GLL algorithms induce a local causal neighborhood, under the distributional assumption of faithfulness and causal sufficiency, employing statistical tests of conditional independence, and preferring to assume a variable is in the local neighborhood whenever a conditional test is not applicable due to small sample (provided that a univariate association exists, otherwise independence is the default) in order to minimize false negative risk of losing a true member and overall risk of false positives and false negatives if true network is not dense. More about the inductive bias of GLL can be found in Aliferis et al. (2010).

3. For networks *Lung_Cancer* and *Gene*, we also add an eleventh target that corresponds to the natural response variable: lung cancer diagnosis and cell cycle state, respectively. For network *Munin* we use only 6 targets because of extreme probability distributions of the majority of variables that do not allow variability in the finite sample of size 500 and even 5000. Because of the same reason, we did not experiment with sample size 200 in the *Munin* network.

We obtained two resimulated networks as follows: (a) *Lung_Cancer* network: We randomly selected 799 genes and a phenotype target (cancer versus normal tissue indicator) from human gene expression data of Bhattacharjee et al. (2001). Then we discretized continuous gene expression data and applied SCA to elicit network structure. (b) *Gene* network: It was obtained from a subset of variables of yeast gene expression data of Spellman et al. (1998) that contained 800 randomly selected genes and a target variable denoting cell cycle state. Continuous gene expression data was also discretized and SCA was applied to learn network. This research design follows Friedman et al. (2000).

Furthermore, we note that additional factors not captured in the simulation or resimulation process make real-life discovery potentially harder than in our experiments. Such factors include for example, deviations of faithfulness, existence of temporal and cellular aggregation effects, unmeasured variables, and various measurement, normalization, and noise artifacts. However evaluations with simulated and resimulated data yield comparative performances that are still highly informative since if a method cannot induce the correct structure from relatively easier settings, it is unlikely that in harder real-life situations it will perform any better. In other words successful causal structure discovery performance in simulated and resimulated networks represents at a minimum "gate-keeper" level performance that will filter the more promising from the less promising methods (Spirtes et al., 2000). Finally, as Spirtes et al. (2000) note the behavior of constraint-based algorithms is particularly complex and theoretical analyses are very difficult to perform. The same is true for several other modern feature selection methods. Hence, simulation experiments are necessary in order to gain a deeper understanding of the strengths and limitations of many state-of-the-art algorithms. The evaluated algorithms are provided in Appendix Table 12.

We evaluate the algorithms using the following metrics:

1. *Graph distance*. This metric calculates the average shortest unoriented graph distance of each variable returned by an algorithm to the local neighborhood of target, normalized by the average such distance of all variables in the graph. The rationale is to normalize the score to allow for comparisons across data sets and to correct the score for randomly choosing variables. The score is a non-negative number and has the following interpretation: value 0 means that current feature set is a subset of the true local neighborhood of the target, values less than 1 are better than random selection in the specific network, values equal to 1 are as good as random selection in the specific network and values higher than 1 are worse than random selection. The metric is computed using Dijkstra's shortest path algorithm.

2. *Euclidean distance from the perfect sensitivity and specificity (in the ROC space)* for discovery of local neighborhood of the target variable. This is computed as in Tsamardinos et al. (2003b) and provides a loss function-neutral combination of sensitivity and specificity.

3. *Proportion of false positives and proportion of false negatives*.

4. *Classification performance using polynomial SVM and KNN classifiers* with parameters optimized by nested cross-validation (misclassification cost C and kernel degree d for SVMs and number of nearest neighbors k for KNN) on an independently sampled test data set with large sample (n=5000). The performance is measured

by AUC (Fawcett, 2003) on binary tasks and proportion of correct classifications on multiclass tasks.

5. *Feature selection time in minutes*. All caveats regarding interpretation of running times stated in Section 5 apply here as well.

We note that the causal discovery evaluations emphasize *local* discovery of direct causes and direct effects and this choice is supported by several reasons. First, in many domains searching for direct causes and effects is natural (e.g., biological pathway discovery). Second, for non-causal feature selection methods, a natural causal interpretation of their output is being among the direct causes and direct effects (or the Markov blanket) of the target. Consider for example clustering or differential gene expression in bioinformatics where if *Gene1* clusters with *Gene2*, or if *Gene3* is more strongly differentially expressed with respect to some phenotype than *Gene4* then *Gene1* and *Gene2* are interpreted to be members of the same pathway (i.e., in close proximity in the gene regulatory/causal network), and *Gene 3* is interpreted to be more likely to determine the phenotype than *Gene4*. Similar interpretations abound for other non-causal feature selection methods. We notice that if a method is locally causally inconsistent then it is very unlikely that it will be globally causally consistent either. The logic of this argument is that algorithms either return global or local causal knowledge. If an algorithm outputs a global causal graph and this is incorrect, then this implies that locally it will be wrong for at least some variables. Conversely, if the global graph is correct then locally it is correct as well. If algorithm B outputs a correct local causal set (e.g., direct causes and direct effects) then we can "piece together" these sets and obtain a correct global graph. Finally, if an algorithm outputs an incorrect non-empty local causal set, this implies that B returns non-causes as direct causes or remote causes as direct causes (and the same for effects). Thus, it is not possible to construct the full causal graph strictly from knowledge provided by the algorithm. As a result, local causal consistency is necessary for global consistency as well.

A second reason for focusing on local causal discovery is that it is much harder in practice than indirect causal discovery in highly interconnected causal networks. In our bioinformatics example, because cancer affects many pathways, it is trivial to find genes affected by cancer, since a large proportion (e.g., half) of the measured genes are expected to be affected. However, it is vastly harder to find the chain of events that leads from occurrence of cancer to *Gene1* becoming under- or over-expressed. In such settings, discovery of remote causation is not particularly hard, neither it is particularly interesting. Conversely, when one has a locally correct causal discovery algorithm as elucidated in Section 2, global causal learners can be relatively easily constructed.

Finally, in our evaluations we do not examine quality of causal orientation of the algorithms output for several reasons: First, while GLL algorithms' output can be oriented by constraint-based or other post-processing, non-causal feature selection methods do not readily admit orientation. Second, orientation is not needed when target T is a terminal variable as is often the case in the real data. Third, oriented local causal discovery is harder than unoriented one (Ramsey et al., 2006), and it makes sense to examine the ability of the feature selection algorithms for causal discovery in tasks of incremental difficulty, especially since as we will see most of the non-causal algorithms do not perform well even when seeking unoriented causality. Fourth, orientation information can be obtained subsequently by experiments or knowledge-based post-processing and in many practical settings it is not the primary obstacle to causal discovery.

6.1. Superiority of Causal Over Non-Causal Feature Selection Methods for Causal Discovery

Causal methods achieve, consistently under a variety of conditions and across all metrics employed, superior causal discovery performance than non-causal feature selection methods in our experiments. Figures 14(a) and 15 compare semi-interleaved HITON-PC to HITON-MB, RFE, UAF, L0, and LARS-EN in terms of graph distance and for different sample sizes. Other GLL instantiations such as Interleaved-HITON-PC, MMPC, and Interleaved-MMPC perform similarly to HITON-PC (data in Table S12 in the online supplement). We apply HITON-PC as is and also with a variable pre-filtering step such that only variables that pass a test of univariate association with the target at 5% False Discovery Rate (FDR) threshold are input into the algorithm (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995). Motivation and analysis of incorporating FDR in GLL is provided in Aliferis et al. (2010).

As can be seen, in all samples HITON-PC variants return features closely localized near the target while HITON-MB requires relatively larger sample size to localize well. The distance is smaller as sample size grows. Methods such as univariate filtering localize features well in some data sets and badly in others. As sample size grows, localization of univariate filtering deteriorates. Methods L0, and LARS-EN exhibit a *reverse-localization* bias (i.e., preferentially select features *away* from the target). Performance of RFE varies greatly across data sets in its ability to localize features and this is independent of sample size. A "bull's eye" plot for *Insurance10* data set is provided in Figure 16. A localization example for *Insurance10* data set is shown in Figure 17. The presented visualization examples are representative of the relative performance of causal versus non-causal algorithms. Table 4 provides p-values (via a permutation test at 5% alpha) for the differences of localization among algorithms.

Tables S13-S16 and Figure S2(a)-(d) in the online supplement compare the same algorithms in terms of (a) Euclidian distance from the point of perfect sensitivity and specificity, (b) proportion of false negatives, (c) proportion of false positives, and (d) running time in minutes. Consistent with the results presented in the main text, local causal discovery algorithms strongly outperform non-causal feature selection methods in ability to find the direct causes and effects of the target variable.

6.2. Classification Performance is Misleading for Causal Discovery

Despite causally wrong outputs (i.e., failing to return the Markov blanket or parents and children set), several non-causal feature selection methods achieve comparable classification performance with causal algorithms in the simulated data. Figure 14(b) (and Tables S17-S18 and Figure S2(e) in the online supplement) shows the average AUC and proportion of correct classifications. This phenomenon is related to information redundancy of features in relation to the target in non-sparse causal processes. In addition, it is facilitated by the relative insensitivity of state-of-the-art classifiers to irrelevant and redundant features. *Good classification performance is thus greatly misleading as a criterion for quality of causal hypotheses* generated by non-causal feature selection algorithms.

In conclusion, the results in the present section strongly undermine the hope that non-causal feature selection methods can be used as good heuristics for causal discovery. The idea that non-causal feature selection can be used for causal discovery should be viewed with caution (Guyon et al., 2007). Whole research programs are, in many domains, built on experiments motivated by causal hypotheses that were generated by

Table 4: Statistical comparison between semi-interleaved HITON-PC with G^2 test (with and w/o FDR correction) and other methods in terms of graph distance. Bolded p-values are statistically significant at 5% alpha.

	Sample size = 200		Sample size = 500		Sample size = 5000	
Comparison	P-value	Nominal winner	P-value	Nominal winner	P-value	Nominal winner
average semi-interleaved HITON-PC with G2 test vs. HITON-MB	**<0.0001**	HITON-PC	**0.0042**	HITON-PC	**0.0472**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. average RFE	0.2594	HITON-PC	**0.0076**	HITON-PC	**<0.0001**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. average UAF	**0.0078**	UAF	0.6788	HITON-PC	**0.0086**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. L0	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC		N/A
average semi-interleaved HITON-PC with G2 test vs. average LARS-EN	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC
average semi-interleaved HITON-PC-FDR with G2 test vs. HITON-MB	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. average RFE	**<0.0001**	HITON-PC-FDR	**0.0028**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. average UAF	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. L0	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR		N/A
average semi-interleaved HITON-PC-FDR with G2 test vs. average LARS-EN	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR

non-causal feature selection results (Zhou et al., 2002; Li et al., 2001; Holmes et al., 2000; Eisen et al., 1998) and this seems an unfortunate and inadvisable practice, in light of existence of principled causal algorithms. On the other hand, generalized local learning algorithms in simulated and resimulated experiments show great potential for local causal discovery.

7. Discussion

In the present section we discuss main findings of this research, state limitations and outline open problems, and give an overview of problems addressed in the companion paper.

7.1. Main Findings

Our experimental evaluation shows that GLL algorithms typically attain the theoretically expected benefits of strong feature set parsimony without loss of performance relative to the best classification attained by any method used in the experiments. The wide range of data sets and algorithms used shows that the sufficient conditions stated in the proofs for correctness for GLL are likely to hold and/or that violations may be small or well tolerated.

The second major result from our experiments is that we showed that use of non-causal feature selection methods for learning causality although very widespread, is generally inadvisable. We used resimulated and simulated data and showed that causally-motivated feature selection methods connect local causal discovery with feature

selection for classification consistent with recent theoretical work. Feature selection algorithms that are not causal have a tendency to return highly predictive feature sets that are scattered all over the network, or that are in the periphery of the network, and cannot be otherwise interpreted in a way that makes useful and consistent causal sense. We strongly caution practitioners to use principled causal discovery algorithms whenever available and to not substitute causal discovery methods with predictive/non-causal feature selection ones for reasons of convenience or due to non familiarity with such methods. Practical software widely exists that can be used to apply state-of-the-art causal methods including the methods studied in the present paper that is available for download from the online supplement.

Finally, the theoretical framework that is based in large part on faithfulness and other assumptions summarized in Sections 2 and 3 is a valuable frame of reference both conceptually and algorithmically. However, we do not consider it to be an absolute and immutable measure by which to judge all new and existing algorithms. Our data shows that algorithms that are not deemed correct under the more general assumptions of the framework (e.g., algorithms that do not employ symmetry correction, or algorithms that use $PC(T)$ instead of $MB(T)$ for feature selection for classification) offer in many real data sets same predictive quality and better computational tractability than the sound algorithms. This is a reflection of several factors. One of them is the existence of distributions that are special classes of faithful ones and are easier to analyze (e.g., where symmetry correction is not required, or in other words where $EPC(T) = PC(T)$). A second factor is mitigating circumstances for violations of assumptions (Aliferis et al., 2010). A third factor is that practical implementations of sound algorithms are statistically imperfect (in other words, a theoretical assumption that conveniently leads to a proof of correctness, for example that a conditional test of independence is correct, does not entail immediate or flawless practical feasibility since all such tests admit errors in practice). An alternative set of assumptions for correctness may require vaguely 'sufficient sample size' disregarding the practical difficulty of determining whether in any given analysis this requirement is met. As a result, practical implementations may claim soundness without being demonstrably sound in applied settings. We address the small-sample behavior of GLL algorithms with empirical analysis in the companion paper (Aliferis et al., 2010).

7.2. Limitations and Open Problems

A possible critique of the present work is that Markov blanket features may not work well with a plethora of classifiers, distributions and loss functions. Indeed, a feature selector that is uniformly optimal is not attainable as shown by the results in Tsamardinos and Aliferis (2003), and several (possibly infinite) conceivable classifiers will fail to capture the information in the selected features. Our focus was to examine if the GLL framework has merit in the sense of whether GLL instantiations when applied and compared to reasonable state-of-the-art baseline feature selectors in many complex data sets from typical analysis domains and with practical classifiers, loss function and sample sizes, yield good performance consistent with the theoretical claims of GLL.

Another possibility we would like to address is that best predictivity achieved in our experiments for each data set may not be optimal since some classifier other than SVMs and KNN may yield better predictivity. We believe that this possibility is remote for the following reason: Evidence from earlier published work where we have applied instances of GLL with classifiers such as ANNs, Decision Trees, Simple Bayes, as well as

SVMs and KNN supports that the choice of classifier matters very little in practice and similar predictivity/parsimony patterns as the ones reported here were found (Aliferis et al., 2003a). On the other hand, the use of SVMs and KNN as classifiers uniformly across our experiments confers many benefits explained in Section 5. To further support the use of these classifiers we provide additional experimental results in Appendix Table 10 where we use features extracted from embedded or wrapper-based feature selectors (L0, RFVS, LARS-EN) and compare SVMs and KNN to classifiers native to the above embedded and wrapper-based methods. We found that SVMs and KNN achieve predictivity comparable to the classifiers from the aforementioned feature selectors.

Additional strong evidence in favor of our conclusions that GLL algorithms yield highly predictive and parsimonious feature sets is given by the simulated and resimulated data experiments where both the data-generative model and optimal feature sets are known. In those experiments the true Markov blanket is directly given by the model and does constitute the gold standard for the smallest and optimally informative feature set for common loss functions in the sense that *it contains all information available for predicting the target*. The experiments showed that the GLL algorithms identify this Markov blanket very well and better than the baseline comparison algorithms.

Although the GLL framework and the studied instantiations and implementations are theoretically well motivated and empirically robust in many practical data analysis domains, as demonstrated in our experiments, as with all machine learning methods they should be expected to not perform well in quality or efficiency in certain distributions. Such distributions may include cases where the Markov blanket is very large and thus the combinatorics of the elimination phase makes it too slow. Another case can be when extreme non-linearities render the $PC(T)$ members "invisible" to the algorithm (because univariate association with the target is zero). Another possibility for hurting efficiency arises when excessive synthesis of information exists such that the true members of $PC(T)$ are not considered before other weakly relevant variables enter the $TPC(T)$. Also when certain types of deterministic relationships exist or more broadly target information equivalence (i.e., special types of violations of faithfulness), many Markov blankets may exist and the algorithms will return a predictively optimal feature set but both causal localization and optimal parsimony may be lost (Statnikov, 2008). The practical importance of these possibilities needs to be assessed domain-by-domain.

Some of the adverse situations described in the limitations sub-section can be addressed by relaxing the algorithm operation (e.g., for very large Markov blankets the analyst can set *max-k* to a very small number and achieve faster execution but incur some false positives). In some domains, violation of assumptions are mitigated by other factors (e.g., Aliferis et al. 2010 describes how connectivity can make extremely epistatic parents visible to the algorithms). These and other situations constitute open research areas and very recent research efforts attempt to address these issues. For example, Statnikov (2008) provides algorithms that address multiplicity of Markov blankets and Tsamardinos and Brown (2008b) introduce a method for kernel mapping of extremely non-linear functions to a faithful feature space that can be used to do feature selection via GLL in the transformed feature space.

Although the emphasis of the present work was in classification, Markov blanket theory applies equally well to regression and thus the GLL framework can be used for regression problems as well. An empirical analysis of performance of regression-oriented GLL instantiations and comparisons to state-of-the-art methods were not pursued here however.

7.3. Further Problems Addressed in the Companion Paper

While the theory motivating local learning and especially Markov blanket induction for feature selection has wide implications, it is far from complete. To begin with, all theoretical arguments to-date apply to the large sample case. While the theory implies that the large-sample Markov blanket and the corresponding classifiers fitted from large sample, are predictively optimal, it is not known to what extend learning from small samples affects the optimality of Markov blanket based feature selection. More specifically, it is not clear how often in small samples and real-life distributions the true Markov blanket (i.e., obtained from the data-generative process) gives an optimal classifier when the latter is fitted from small samples with state-of-the-art classifiers. Similarly, we do not know whether the estimated Markov blanket gives an optimal classifier when the latter is fitted from small samples or even when it is fitted from the large sample. Related to the above for practical applications, we do not know how fast is convergence of the estimated Markov blanket/classifier to true Markov blanket/optimal classification as a function of sample size, for the available state-of-the-art Markov blanket inducing algorithms. In the second part of our work (Aliferis et al., 2010) we examine these issues. We also provide explanations why counter-intuitively relaxed versions of some algorithms that trade-off computational efficiency for theoretical soundness tend to outperform sound versions in some domains. Moreover, we systematically study the factors that influence the quality and number of statistical decisions, explain the inductive bias of the algorithms, show how non-causal feature selection methods can be understood in light of Markov blanket induction theory, and address divide-and-conquer local to global causal graph learning strategies.

Appendix A.

This Appendix provides proofs of theorems and additional tables referenced in the paper.

A.1. Proof of Theorem 2

Consider the algorithm in Figure 4. First notice, that as we mentioned above, when conditions (a) and (c) hold the direct causes and direct effects of T will coincide with the parents and children of T in the causal Bayesian network G that faithfully captures the distribution (Spirtes et al., 2000). As we have shown in Section 4 and in Tsamardinos et al. (2003b), the $PC_G(T) = PC(T)$ is unique in all networks faithfully capturing the distribution.

First we show that the algorithm will terminate, that is that the termination criterion of admissibility rule #3 will be met. The criterion requires that no variable eligible for inclusion will fail to enter $TPC(T)$ and that no variable that can be eliminated from $TPC(T)$ is left inside. Indeed because (a) due to admissibility rule #1 all eligible variables in OPEN are identified, (b) $\boldsymbol{V}$ is finite and OPEN instantiated to $\boldsymbol{V} \setminus \{T\}$, and (c) termination will not happen before all eligible members of OPEN are moved from OPEN to $TPC(T)$, the first part of the termination criterion will be satisfied. The second part of the termination criterion will also be satisfied because of admissibility rule #2 which examines for removal all variables and discards the ones that can be removed.

Lemma 1 *The output of GLL-PC-nonsym $TPC(T)$ is such that: $PC(T) \subseteq TPC(T) \subseteq EPC(T)$.*

Proof Let us assume that $X \in PC(T)$ and show that $X \in TPC(T)$ by the end of GLL-PC-nonsym. By admissibility rule #3, X will never fail to enter $TPC(T)$ by the end of GLL-PC-nonsym. By Theorem 1, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X\}$, $\neg I(X, T \mid \boldsymbol{Z})$ and so the elimination strategy because of admissibility rule #2 will never remove X from $TPC(T)$ by the end of GLL-PC-nonsym.

Now, let us assume that $X \in TPC(T)$ by the end of GLL-PC-nonsym and show that $X \in EPC(T)$. Let us assume the opposite, that is, that $X \notin EPC(T)$ and so by definition $I(X, T \mid \boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$. By the same argument as in the previous paragraph, we know that at some point before termination of the algorithm, in step 4, $TPC(T)$ will contain the $PC(T)$. Since $X \notin EPC(T)$, the elimination strategy will find that $I(X, T \mid \boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$ and remove X from $TPC(T)$ contrary to what we assumed. Thus, $X \in EPC(T)$ by the end of GLL-PC-nonsym. ■

Lemma 2 *If $X \in EPC(T) \setminus PC(T)$, then $T \notin EPC(X) \setminus PC(X)$*

Proof Let us assume that $X \in EPC(T) \setminus PC(T)$. For every network G faithful to the distribution P $Parents_G(T) \subseteq PC_G(T) = PC(T)$. X has to be a descendant of T in every network G faithful to the distribution because if it is not a descendant, then there is a subset $\boldsymbol{Z}$ of T's parents s.t., $I(X, T \mid \boldsymbol{Z})$ (by the Markov Condition). Since $X \in EPC(T) \setminus PC(T)$, we know that by definition $\neg I(X, T \mid \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$. By the same argument, if also $T \in EPC(X) \setminus PC(X)$, T would have to be a descendant of X in the every network G which is impossible since the networks are acyclic. So, $T \notin EPC(X) \setminus PC(X)$. ■

Let us assume that $X \in PC(T)$. By Lemma 1, $X \in TPC(T)$ by the end of GLL-PC-nonsym. Since also $T \in PC(X)$, substituting X for T, we also have that by the end of GLL-PC-nonsym, $T \in TPC(X)$. So, X will not be removed from $\boldsymbol{U}$ by the symmetry requirement of GLL-PC either, and will be in the final output of the algorithm.

Conversely, let us assume that $X \notin PC(T)$ and show $X \notin \boldsymbol{U}$ at termination of algorithm GLL-PC. If X never enters $TPC(T)$ by the inclusion heuristic, the proof is done. Similarly, if X enters but is later removed from $TPC(T)$ by the exclusion strategy, the proof is done too. So, let us assume that X enters $TPC(T)$ at some point and by the end of GLL-PC-nonsym(T) is not removed by the exclusion strategy. By Lemma 1, we get that by the end of GLL-PC-nonsym, $X \in EPC(T)$ and since we assumed $X \notin PC(T)$, we get that $X \in EPC(T) \setminus PC(T)$. By Lemma 2, we get that $T \notin EPC(X) \setminus PC(X)$. Since also $T \notin PC(X)$, we get that $T \notin EPC(X)$. Step 3 of GLL-PC will thus eliminate X from $\boldsymbol{U}$.

A.2. Proof of Theorem 4

Since we assume faithful Bayesian networks, d-separation in the graph of such a network is equivalent to independence and can be used interchangeably (Spirtes et al., 2000).

If $X \in MB(T)$, we show $X \in TMB(T)$ in the end. If $X \in MB(T)$ and $X \in PC(T)$, it will be included in the $TMB(T)$ in step 3, will not be removed afterwards and will be included in the final output.

If $X \in MB(T) \setminus PC(T)$ then X will be included in $\boldsymbol{S}$ since if X is a spouse of T, there exists Y (by definition of spouse) s.t., $X \in PC(Y)$, $Y \in PC(T)$ and $X \notin PC(T)$. For that

Y, by Theorem 3 we know that $\neg I(X, T \mid \boldsymbol{Z} \cup \{Y\})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ and so the test in step 5c will succeed and X will be included in $TMB(T)$ in the end.

Conversely, if $X \notin MB(T)$ we show that $X \notin TMB(T)$ by the end of the algorithm. Let $\boldsymbol{Z}$ be the subset in step 5a, s.t., $I(X, T \mid \boldsymbol{Z})$ (i.e., $\boldsymbol{Z}$ d-separates X and T). Then, $\boldsymbol{Z}$ blocks all paths from X to T. For the test in step 5c to succeed a node Y must exist that opens a new path, previously closed by $\boldsymbol{Z}$, from X to T. Since by conditioning on an additional node a path opens, Y has to be a collider (by the d-separation definition) or a descendant of a collider on a path from X to T. In addition, this path must have length two edges since all nodes in $\boldsymbol{S}$ are the parents and children of the $PC(T)$ but without belonging in $PC(T)$. Thus, for the test in step 5c to succeed there has to be a path of length two from X to T with a collider in-between, that is, X has to be a spouse of T. Since $X \notin MB(T)$ the test will fail for all Y and $X \notin TMB(T)$ by the end of the algorithm.

References

C. F. Aliferis and G. F. Cooper. An evaluation of an algorithm for inductive learning of Bayesian belief networks using simulated data sets. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1994.

C. F. Aliferis and I. Tsamardinos. Algorithms for large-scale local causal discovery and feature selection in the presence of small sample or large causal neighborhoods. *Technical Report DSL 02-08*, 2002a.

C. F. Aliferis and I. Tsamardinos. Using local causal induction to improve global causal discovery: Enhancing the sparse candidate set. *Technical Report DSL 02-04*, 2002b.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using Markov blanket induction for the prediction of protein-drug binding. *Technical Report DSL 02-06*, 2002.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: a novel Markov blanket algorithm for optimal variable selection. *AMIA 2003 Annual Symposium Proceedings*, pages 21–25, 2003a.

C. F. Aliferis, I. Tsamardinos, A. Statnikov, and L. E. Brown. Causal explorer: a causal probabilistic network learning toolkit for biomedical discovery. *Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*, 2003b.

C. F. Aliferis, A. Statnikov, E. Kokkotou, P. P. Massion, and I. Tsamardinos. Local regulatory-network inducing algorithms for biomarker discovery from mass-throughput datasets. *Technical Report DSL 06-05*, 2006a.

C. F. Aliferis, A. Statnikov, and P. P. Massion. Pathway induction and high-fidelity simulation for molecular signature and biomarker discovery in lung cancer using microarray gene expression data. *Proceedings of the 2006 American Physiological Society Conference "Physiological Genomics and Proteomics of Lung Disease"*, 2006b.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification. Part II: Analysis and extensions. *Journal of Machine Learning Research*, 11:235–284, 2010.

Y. Aphinyanaphongs and C. F. Aliferis. Learning boolean queries for article quality filtering. *Medinfo 2004.*, 11(Pt 1):263–267, 2004.

Y. Aphinyanaphongs, A. Statnikov, and C. F. Aliferis. A comparison of citation metrics to machine learning filters for the identification of high quality medline documents. *J.Am.Med.Inform.Assoc.*, 13(4):446–455, Jul 2006.

X. Bai, C. Glymour, R. Padman, J. Ramsey, P. Spirtes, and F. Wimberly. PCX: Markov blanket classification for large data sets with few cases. *Technical Report, Center for Automated Learning and Discovery*, 2004.

G. E. A. P. A. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society.Series B (Methodological)*, 57(1):289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann.Statist*, 29(4):1165–1188, 2001.

A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc.Natl.Acad.Sci.U.S.A*, 98(24):13790–13795, Nov 2001.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

L. E. Brown, I. Tsamardinos, and C. F. Aliferis. A comparison of novel and state-of-the-art polynomial Bayesian network learning algorithms. *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

R. Caruana and D. Freitag. Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, 1994.

J. Cheng and R. Greiner. Comparing Bayesian network classifiers. *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 101–107, 1999.

J. Cheng and R. Greiner. Learning Bayesian belief network classifiers: Algorithms and system. *Proceedings of 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, 2001.

J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137(1):43–90, 2002a.

J. Cheng, C. Hatzis, H. Hayashi, M. A. Krogel, S. Morishita, D. Page, and J. Sese. Kdd cup 2001 report. *ACM SIGKDD Explorations Newsletter*, 3(2):47–64, 2002b.

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

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(3):507–554, 2003.

D. M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks is NP-hard. *Technical Report MSR-TR-94-17*, 1994.

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

T. P. Conrads, V. A. Fusaro, S. Ross, D. Johann, V. Rajapakse, B. A. Hitt, S. M. Steinberg, E. C. Kohn, D. A. Fishman, G. Whitely, J. C. Barrett, L. A. Liotta, E. F. I. I. I. Petricoin, and T. D. Veenstra. High-resolution serum proteomic features for ovarian cancer detection. *Endocr.Relat Cancer*, 11(2):163–178, Jun 2004.

G. F. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1(2):203–224, 1997.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

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

G. F. Cooper, C. F. Aliferis, R. Ambrosino, J. Aronis, B. G. Buchanan, R. Caruana, M. J. Fine, C. Glymour, G. Gordon, and B. H. Hanusa. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9(2): 107–138, 1997.

D. Dash and G. F. Cooper. Exact model averaging with naive Bayesian classifiers. *Proc.19th Int.Conf.Machine Learning (ICML 2002)*, pages 91–98, 2002.

E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3):837–845, Sep 1988.

R. Diaz-Uriarte and S. Alvarez de Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

S. Duda, C. F. Aliferis, R. Miller, A. Statnikov, and K. Johnson. Extracting drug-drug interaction articles from medline to improve the content of drug databases. *AMIA 2005 Annual Symposium Proceedings*, pages 216–220, 2005.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in model selection and performance assessment. *UC Berkeley Division of Biostatistics Working Paper Series*, 126, 2003.

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

F. Eberhardt, C. Glymour, and R. Scheines. N-1 experiments suffice to determine the causal relations among n variables. *Innovations in Machine Learning: Theory And Applications*, 2006.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc.Natl.Acad.Sci.U.S.A*, 95(25):14863–14868, Dec 1998.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1361, 2001.

R. E. Fan, P. H. Chen, and C. J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6(1889): 1918, 2005.

N. Fananapazir, M. Li, D. Spentzos, and C. F. Aliferis. Formative evaluation of a prototype system for automated analysis of mass spectrometry data. *AMIA 2005 Annual Symposium Proceedings*, pages 241–245, 2005.

T. Fawcett. Roc graphs: Notes and practical considerations for researchers. *Technical Report, HPL-2003-4, HP Laboratories*, 2003.

D. P. Foster and R. A. Stine. Variable selecion in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, 99(466):303–314, 2004.

L. Frey, D. Fisher, I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Identifying Markov blankets with decision tree induction. *Proceedings of the Third IEEE International Conference on Data Mining (ICDM)*, 2003.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, pages 206–215, 1999a.

N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: the "sparse candidate" algorithm. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999b.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J Comput.Biol.*, 7(3-4):601–620, 2000.

G. M. Fung and O. L. Mangasarian. A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, 28(2):185–202, 2004.

T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, Oct 2000.

O. Gevaert, Smet F. De, D. Timmerman, Y. Moreau, and Moor B. De. Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. *Bioinformatics*, 22(14):e184–e190, Jul 2006.

C. N. Glymour and G. F. Cooper. *Computation, Causation, and Discovery*. AAAI Press, Menlo Park, Calif, 1999.

P. I. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, volume 2nd. Springer, New York, 2000.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(1):1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications*. Springer-Verlag, Berlin, 2006a.

I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider, and M. Uhr. Feature selection with the clop package. Technical report, http://clopinet.com/isabelle/Projects/ETH/TM-fextract-class.pdf, 2006b.

I. Guyon, C. F. Aliferis, and A. Elisseeff. *Computational Methods of Feature Selection*, chapter Causal Feature Selection. Chapman and Hall, 2007.

D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear SVM-based feature selection. *Proceedings of the Twenty First International Conference on Machine Learning (ICML)*, 2004.

D. Heckerman. A tutorial on learning with Bayesian networks. *Technical Report MSR-TR-95-06*, 1995.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

M. Hollander and D. Wolfe. *Nonparametric Statistical Methods*, volume 2nd. Wiley, New York, NY, USA, 1999.

J. H. Holmes, D. R. Durbin, and F. K. Winston. The learning classifier system: an evolutionary computation approach to knowledge discovery in epidemiologic surveillance. *Artif.Intell.Med.*, 19(1):53–74, May 2000.

N. Hoot, I. Feurer, C. W. Pinson, and C. F. Aliferis. Modelling liver transplant survival: comparing techniques of deriving predictor sets. *Journal of Gastrointestinal Surgery*, 9 (4):563, Apr 2005.

T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, Boston, 2002.

K. Kira and L. A. Rendell. A practical approach to feature selection. *Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256, 1992.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

D. Koller and M. Sahami. Toward optimal feature selection. *Proceedings of the International Conference on Machine Learning*, 1996, 1996.

I. Kononenko. Estimating attributes: Analysis and extensions of relief. *Proceedings of the European Conference on Machine Learning*, pages 171–182, 1994.

L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17(12):1131–1142, Dec 2001.

H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic, Boston, 1998.

H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: an enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

S. Mani and G. F. Cooper. A study in causal discovery from population-based infant birth and death records. *Proceedings of the AMIA Annual Fall Symposium*, 319, 1999.

S. Mani and G. F. Cooper. Causal discovery using a Bayesian local causal discovery algorithm. *Medinfo 2004.*, 11(Pt 1):731–735, 2004.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. *Advances in Neural Information Processing Systems*, 12:505–511, 1999.

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

A. Moore and W. K. Wong. Optimal reinsertion: a new search operator for accelerated and more accurate Bayesian network structure learning. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 552–559, 2003.

K. P. Murphy. Active learning of causal Bayes net structure. *Technical Report, University of California, Berkeley*, 2001.

R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. Wiley, New York, 1990.

R. E. Neapolitan. *Learning Bayesian networks*. Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

J. Peña, J. Bjorkegren, and J. Tegner. Growing Bayesian network models of gene networks from seed genes. *Bioinformatics*, 21(2):224–229, 2005a.

J. Peña, J. Bjorkegren, and J. Tegner. Scalable, efficient and correct learning of Markov boundaries under the faithfulness assumption. *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2005b.

J. Peña, R. Nilsson, J. Bjorkegren, and J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45(2): 211–232, 2007.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, U.K, 2000.

J. Pearl and T. Verma. A theory of inferred causation. *Principles of Knowledge Representation and Reasoning: Proceedings of Second International Conference*, pages 441–452, 1991.

J. Pearl and T. S. Verma. Equivalence and synthesis of causal models. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

I. Pournara and L. Wernisch. Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20(17):2934–2942, Nov 2004.

A. Rakotomamonjy. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3(7-8):1357–1370, 2003.

J. Ramsey. A pc-style Markov blanket search for high-dimensional datasets. *Technical Report, CMU-PHIL-177, Carnegie Mellon University, Department of Philosophy*, 2006.

J. Ramsey, J. Zhang, and P. Spirtes. Adjacency-faithfulness and conservative causal inference. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 2006.

A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. Lopez-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *N.Engl.J Med.*, 346(25):1937–1947, Jun 2002.

A. Sboner and C. F. Aliferis. Modeling clinical judgment and implicit guideline compliance in the diagnosis of melanomas using machine learning. *AMIA 2005 Annual Symposium Proceedings*, pages 664–668, 2005.

T. Scheffer. *Error Estimation and Model Selection*. PhD thesis, Ph.D.Thesis, Technischen Universitet Berlin, School of Computer Science, 1999.

C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 4(2):163–192, 2000.

P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol.Biol Cell*, 9(12):3273–3297, Dec 1998.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*, volume 2nd. MIT Press, Cambridge, Mass, 2000.

A. Statnikov. Algorithms for discovery of multiple Markov boundaries: Application to the molecular signature multiplicity problem. *Ph.D.Thesis, Department of Biomedical Informatics, Vanderbilt University*, 2008.

A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643, Mar 2005a.

A. Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis. Gems: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int.J.Med.Inform.*, 74(7-8):491–503, Aug 2005b.

A. Statnikov, D. Hardin, and C. F. Aliferis. Using SVM weight-based methods to identify causally relevant and non-causally relevant variables. *Proceedings of the NIPS 2006 Workshop on Causality and Feature Selection*, 2006.

J. Tian and J. Pearl. Causal discovery from changes: A bayesian approach. *UCLA Cognitive Systems Laboratory, Technical Report (R-285)*, 2001.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society.Series B (Methodological)*, 58(1):267–288, 1996.

S. Tong and D. Koller. Active learning for structure in bayesian networks. *Proceedings of the International Joint Conference on Artificial Intelligence*, 17:863–869, 2001.

I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AI & Stats)*, 2003.

I. Tsamardinos and L. E. Brown. Bounding the false discovery rate in local Bayesian network learning. *Proceedings of the Twenty Third National Conference on Artificial Intelligence (AAAI)*, 2008a.

I. Tsamardinos and L. E. Brown. Markov blanket-based variable selection in feature space. *Technical report DSL-08-01*, 2008b.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale Markov blanket discovery. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 376–381, 2003a.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 673–678, 2003b.

I. Tsamardinos, C. F. Aliferis, A. Statnikov, and L. E. Brown. Scaling-up Bayesian network learning to thousands of variables using local learning technique. *Technical Report DSL 03-02*, 12, 2003c.

I. Tsamardinos, Brown L.E., and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Technical report DSL-05-01*, 2005.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

L. Wang, J. Zhu, and H. Zou. The doubly regularized support vector machine. *Statistica Sinica*, 16:589–615, 2006.

Y. Wang, J. G. Klijn, Y. Zhang, A. M. Sieuwerts, M. P. Look, F. Yang, D. Talantov, M. Timmermans, M. E. Meijer-van Gelder, J. Yu, T. Jatkoe, E. M. Berns, D. Atkins, and J. A. Foekens. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365(9460):671–679, Feb 2005.

J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3(7):1439–1461, 2003.

S. Yaramakala and D. Margaritis. Speculative Markov blanket discovery for optimal feature selection. *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 809–812, 2005.

C. Yoo and G. F. Cooper. An evaluation of a system that recommends microarray experiments to perform to discover gene-regulation pathways. *Artif.Intell.Med.*, 31(2): 169–182, Jun 2004.

X. Zhou, M. C. J. Kao, and W. H. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences*, 99(20):12783–12788, 2002.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. *Advances in Neural Information Processing Systems (NIPS)*, 16, 2004.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B(Statistical Methodology)*, 67(2):301–320, 2005.

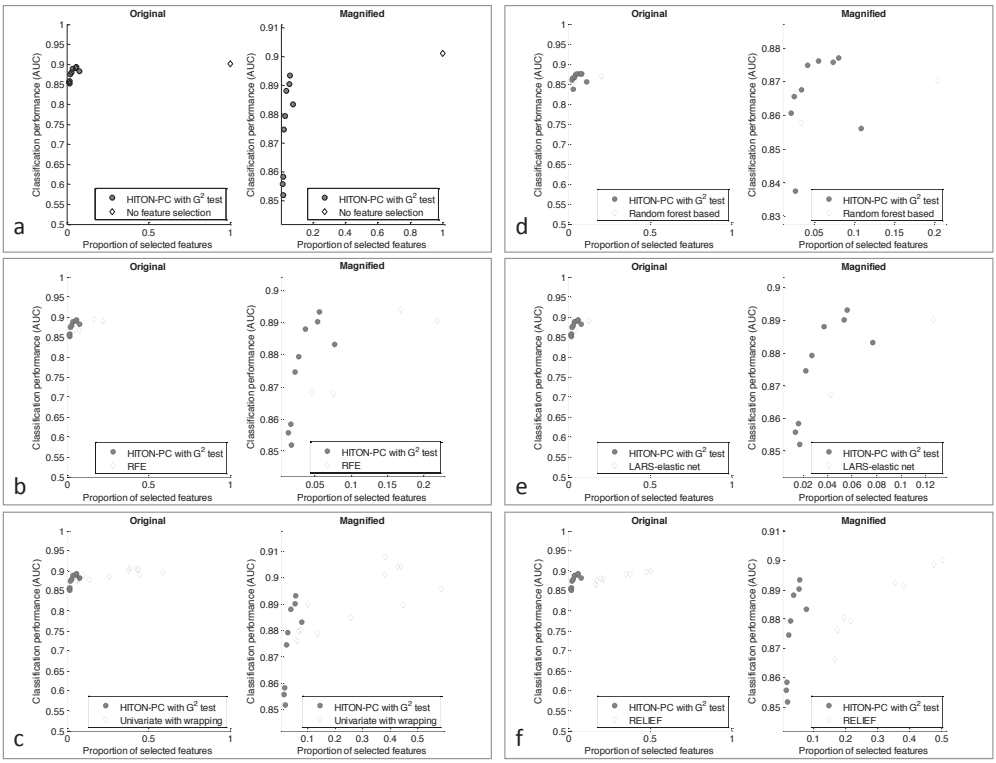


Figure 11: Causal Feature Selection Returns More Compact Feature Sets Than Non-Causal Feature Selection—Comparison of each algorithmic family with semi-interleaved HITON-PC with G^2 test. HITON-PC is executed with 9 different configurations: $\{max\text{-}k = 1, \alpha = 0.05\}$, $\{max\text{-}k = 2, \alpha = 0.05\}$, $\{max\text{-}k = 3, \alpha = 0.05\}$, $\{max\text{-}k = 4, \alpha = 0.05\}$, $\{max\text{-}k = 1, \alpha = 0.01\}$, $\{max\text{-}k = 2, \alpha = 0.01\}$, $\{max\text{-}k = 3, \alpha = 0.01\}$, $\{max\text{-}k = 4, \alpha = 0.01\}$, and a configuration that selects one of the above parameterizations by nested cross-validation. Results shown are averaged across all real data sets where both HITON-PC with G^2 test and an algorithmic family under consideration are applicable and terminate within 2 days of single-CPU time per run on a single training set. Multiple points for each algorithm correspond to different parameterizations/configurations. See Appendix Tables 5–7 for detailed list of algorithms. The left graph has x-axis (proportion of selected features) ranging from 0 to 1 and y-axis (classification performance AUC) ranging from 0.5 to 1. The right graph has the same data, but the axes are magnified to see the details better. This figure is continued in Figures 12 and 13.

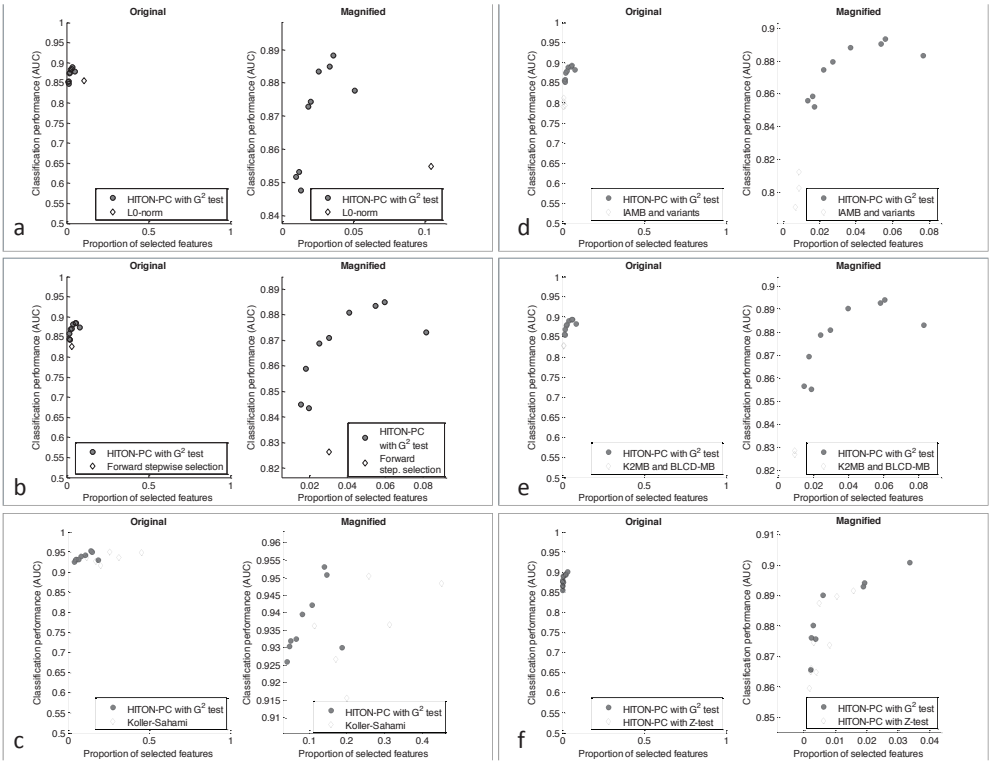


Figure 12: Continued from Figure 11.

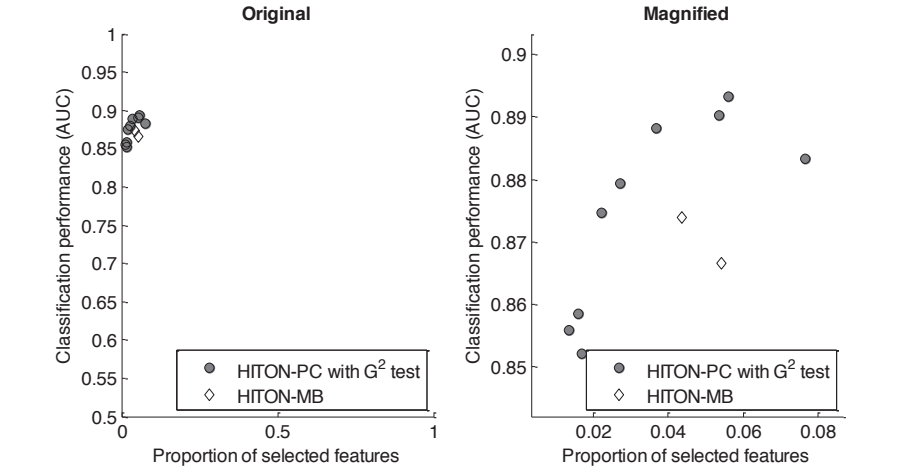


Figure 13: Continued from Figure 12.

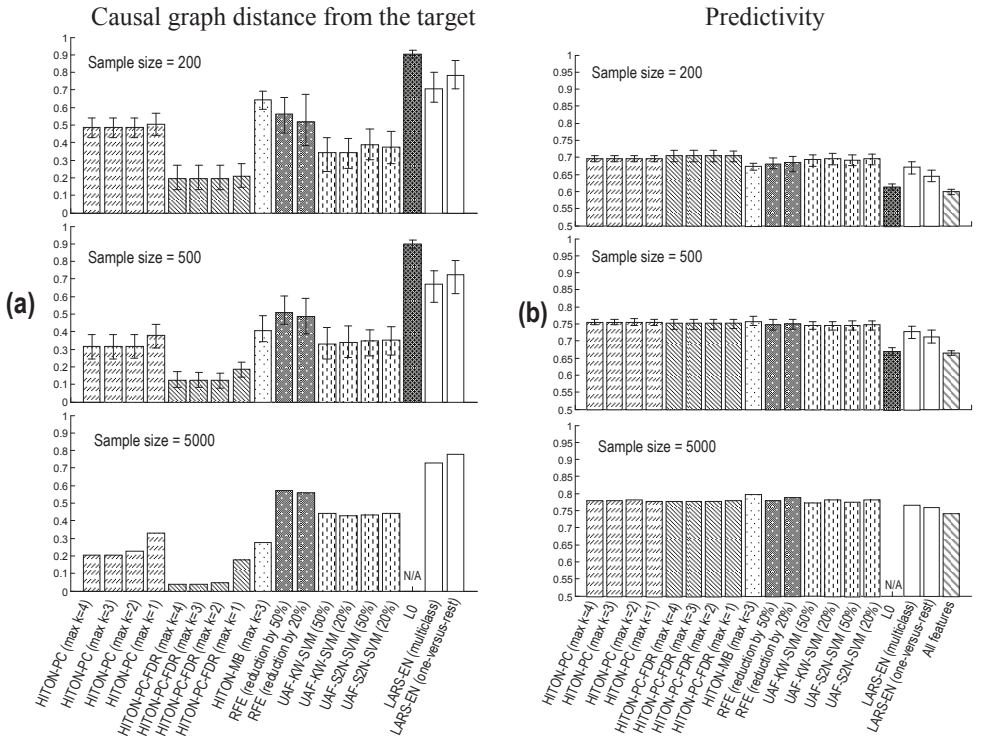


Figure 14: Performance of feature selection algorithms in 9 simulated and resimulated data sets: (a) *graph distance*, (b) *classification performance of polynomial SVM classifiers*. The smaller is causal graph distance and the larger is classification performance, the better is the algorithm. The results are given for training sample sizes = 200, 500, and 5000. The bars denote maximum and minimum performance over multiple training samples of each size (data is available only for sample sizes 200 and 500). The metrics reported in the figure are averaged over all data sets, selected targets, and multiple samples of each size. L0 did not terminate within 2 days (per target) for sample size 5000. Please see text for more details.

Sample size 200

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=3)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=2)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=1)	0.45	0.42	0.42	0.83	0.41	0.46	0.53	0.50	0.50
HITON-PC-FDR (max k=4)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=3)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=2)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=1)	0.29	0.15	0.24	0.18	0.10	0.17	0.34	0.18	0.21
HITON-MB (max k=3)	0.70	0.68	0.50	0.99	0.49	0.66	0.50	0.64	0.64
RFE (reduction of features by 50%)	0.58	0.38	0.50	0.71	0.52	0.45	0.75	0.59	0.56
RFE (reduction of features by 20%)	0.57	0.46	0.54	0.65	0.46	0.30	0.63	0.54	0.52
UAF-KruskalWallis-SVM (50%)	0.45	0.27	0.32	0.50	0.26	0.34	0.34	0.26	0.34
UAF-KruskalWallis-SVM (20%)	0.43	0.32	0.38	0.55	0.27	0.29	0.29	0.22	0.34
UAF-Signal2Noise-SVM (50%)	0.47	0.31	0.44	0.47	0.33	0.35	0.46	0.27	0.39
UAF-Signal2Noise-SVM (20%)	0.44	0.35	0.40	0.56	0.28	0.29	0.44	0.25	0.38
L0	0.95	0.93	0.83	0.97	0.99	0.83	0.82	0.92	0.90
LARS-EN (for multiclass response)	0.67	0.70	0.64	0.79	0.78	0.66	0.64	0.78	0.71
LARS-EN (one-versus-rest)	0.83	0.68	0.67	0.92	0.89	0.70	0.67	0.89	0.78

Sample size 500

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Munin	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.28	0.32	0.31
HITON-PC (max k=3)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.28	0.32	0.31
HITON-PC (max k=2)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.29	0.32	0.32
HITON-PC (max k=1)	0.24	0.28	0.37	0.57	0.34	0.39	0.24	0.52	0.45	0.38
HITON-PC-FDR (max k=4)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.14	0.07	0.12
HITON-PC-FDR (max k=3)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.13	0.07	0.12
HITON-PC-FDR (max k=2)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.11	0.07	0.12
HITON-PC-FDR (max k=1)	0.09	0.11	0.23	0.13	0.08	0.12	0.29	0.40	0.22	0.19
HITON-MB (max k=3)	0.28	0.34	0.37	0.85	0.30	0.43	0.35	0.34	0.38	0.41
RFE (reduction of features by 50%)	0.63	0.51	0.61	0.53	0.37	0.40	0.26	0.70	0.56	0.51
RFE (reduction of features by 20%)	0.54	0.48	0.69	0.53	0.41	0.39	0.26	0.58	0.49	0.49
UAF-KruskalWallis-SVM (50%)	0.37	0.27	0.42	0.49	0.21	0.39	0.34	0.27	0.24	0.33
UAF-KruskalWallis-SVM (20%)	0.40	0.27	0.41	0.48	0.26	0.40	0.30	0.26	0.25	0.34
UAF-Signal2Noise-SVM (50%)	0.40	0.27	0.42	0.51	0.22	0.45	0.29	0.33	0.22	0.35
UAF-Signal2Noise-SVM (20%)	0.42	0.30	0.43	0.51	0.23	0.43	0.30	0.32	0.24	0.35
L0	0.98	0.97	0.93	0.98	0.99	0.87	0.53	0.87	0.97	0.90
LARS-EN (for multiclass response)	0.67	0.71	0.70	0.75	0.78	0.68	0.33	0.60	0.79	0.67
LARS-EN (one-versus-rest)	0.70	0.74	0.74	0.91	0.90	0.77	0.30	0.62	0.82	0.72

Sample size 5000

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Munin	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.13	0.16	0.25	0.35	0.20	0.19	0.04	0.23	0.30	0.20
HITON-PC (max k=3)	0.13	0.16	0.25	0.35	0.20	0.19	0.04	0.23	0.30	0.20
HITON-PC (max k=2)	0.13	0.17	0.25	0.33	0.22	0.19	0.04	0.36	0.33	0.23
HITON-PC (max k=1)	0.18	0.27	0.29	0.33	0.30	0.42	0.04	0.63	0.50	0.33
HITON-PC-FDR (max k=4)	0.00	0.03	0.10	0.10	0.00	0.08	0.04	0.00	0.00	0.04
HITON-PC-FDR (max k=3)	0.00	0.03	0.10	0.10	0.00	0.08	0.04	0.00	0.00	0.04
HITON-PC-FDR (max k=2)	0.00	0.05	0.10	0.10	0.00	0.08	0.04	0.08	0.00	0.05
HITON-PC-FDR (max k=1)	0.01	0.17	0.14	0.11	0.16	0.16	0.04	0.55	0.23	0.18
HITON-MB (max k=3)	0.17	0.20	0.28	0.38	0.27	0.30	0.20	0.33	0.35	0.28
RFE (reduction of features by 50%)	0.63	0.64	0.58	0.59	0.40	0.90	0.28	0.66	0.48	0.57
RFE (reduction of features by 20%)	0.58	0.58	0.69	0.54	0.54	0.92	0.22	0.50	0.43	0.56
UAF-KruskalWallis-SVM (50%)	0.37	0.37	0.62	0.55	0.42	0.69	0.38	0.39	0.20	0.44
UAF-KruskalWallis-SVM (20%)	0.37	0.40	0.60	0.54	0.27	0.59	0.41	0.42	0.24	0.43
UAF-Signal2Noise-SVM (50%)	0.46	0.35	0.65	0.54	0.43	0.67	0.24	0.31	0.25	0.43
UAF-Signal2Noise-SVM (20%)	0.39	0.42	0.58	0.51	0.31	0.60	0.39	0.50	0.25	0.44
LARS-EN (for multiclass response)	0.67	0.85	0.65	0.87	0.74	0.75	0.52	0.71	0.79	0.73
LARS-EN (one-versus-rest)	0.71	0.86	0.74	0.84	0.95	0.80	0.48	0.74	0.88	0.78

Figure 15: Causal graph distance results for training sample sizes = 200, 500 and 5000. The results reported in the figure are averaged over all selected targets. Lighter cells correspond to smaller (better) values of graph distance; darker cells correspond to larger (worse) values of graph distance. L0 did not terminate within 2 days (per target) for sample size 5000.

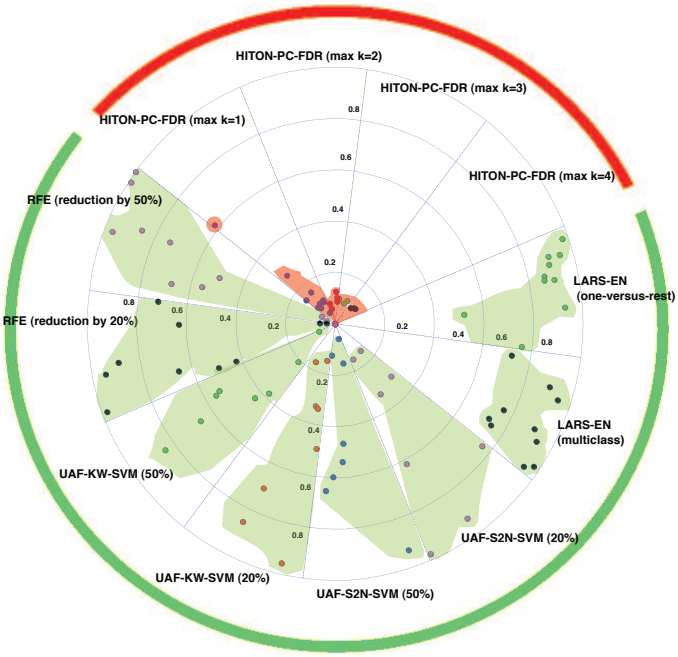


Figure 16: Visualization of graph distances for *Insurance10* network and sample size 5000 by "bull's eye" plot. For each method, results for 10 randomly selected targets are shown. The closer are points to the origin, the better is ability for local causal discovery. Results for GLL method HITON-PC-FDR are highlighted with red; results for baseline methods are highlighted with green.

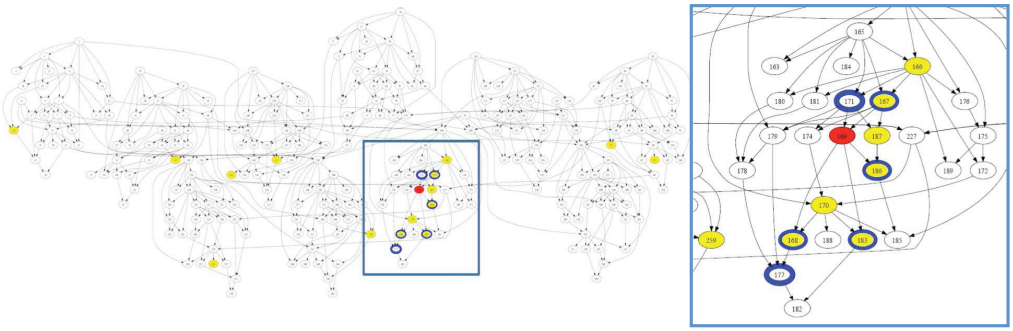


Figure 17: An example of poor localization by a baseline method and good localization by a GLL method. *Left*: Graph of the adjacency matrix of *Insurance10* network. Target variable is shown with red. HITON-PC discovers all 5 members of the parents and children set and a false positive variable #177 that is located close to the true neighborhood (discovered variables are shown with blue bolded circles). RFE discovers 4 out of 5 members of the PC set and introduces many false positives scattered throughout the network (discovered variables are shown with yellow circles). *Right*: A magnified area of the *Insurance10* network close to the target variable.

Table 5: Algorithms used in evaluation on real data sets. When statistical comparison was performed inside a wrapper, we used a non-parametric method by DeLong et al. (1988). The only exception is Random Forest-based Variable Selection (RFVS), where we used a method recommended by its authors (Diaz-Uriarte and Alvarez de Andres, 2006). For GLL algorithms (i.e., variants of HITON-PC, HITON-MB, MMPC, MMMB) we experimented with both G^2 and Fisher's Z-test whenever the latter was applicable. This table is continued in Tables 6 and 7.

Method	**Additional Information**	**Reference**
No feature selection		
RFE (recursive feature elimination SVM-based method)	• reduction by 50% at each iteration, best performing feature subset is returned • reduction by 20% at each iteration, best performing feature subset is returned • reduction by 50% at each iteration, statistically same as best performing feature subset is returned • reduction by 20% at each iteration, statistically same as best performing feature subset is returned	(Guyon et al., 2002)
UAF-KruskalWallis-SVM (univariate ranking by Kruskal-Wallis statistic and feature selection with SVM backward wrapper)	• reduction by 50% at each iteration, best performing feature subset is returned • reduction by 20% at each iteration, best performing feature subset is returned • reduction by 50% at each iteration, statistically same as best performing feature subset is returned • reduction by 20% at each iteration, statistically same as best performing feature subset is returned	(Statnikov et al., 2005a; Hollander and Wolfe, 1999)
UAF-Signal2Noise-SVM (univariate ranking by signal-to-noise statistic and feature selection with SVM backward wrapper)	• reduction by 50% at each iteration, best performing feature subset is returned • reduction by 20% at each iteration, best performing feature subset is returned • reduction by 50% at each iteration, statistically same as best performing feature subset is returned • reduction by 20% at each iteration, statistically same as best performing feature subset is returned	(Guyon et al., 2006b; Statnikov et al., 2005a; Furey et al., 2000)
UAF-Neal-SVM (univariate ranking by Radford Neal's statistic and feature selection with SVM backward wrapper)	• reduction by 50% at each iteration, best performing feature subset is returned • reduction by 20% at each iteration, best performing feature subset is returned • reduction by 50% at each iteration, statistically same as best performing feature subset is returned • reduction by 20% at each iteration, statistically same as best performing feature subset is returned	Chapter 10 in Guyon et al. (2006a)
Random Forest Variable Selection (RFVS)	• best performing feature subset is returned • statistically same as best performing feature subset is returned	(Diaz-Uriarte and Alvarez de Andres, 2006; Breiman, 2001)

Table 6: Continued from Table 5.

Method	Additional Information	Reference
LARS-Elastic Net (LARS-EN)	• best performing feature subset is returned • statistically same as best performing feature subset is returned	(Zou and Hastie, 2005)
RELIEF (with backward wrapping by SVM)	• Number of neighbors = 1, reduction by 50% at each iteration, best performing feature subset is returned • Number of neighbors = 1, reduction by 20% at each iteration, best performing feature subset is returned • Number of neighbors = 5, reduction by 50% at each iteration, best performing feature subset is returned • Number of neighbors = 5, reduction by 20% at each iteration, best performing feature subset is returned • Number of neighbors = 1, reduction by 50% at each iteration, statistically same as best performing feature subset is returned • Number of neighbors = 1, reduction by 20% at each iteration, statistically same as best performing feature subset is returned • Number of neighbors = 5, reduction by 50% at each iteration, statistically same as best performing feature subset is returned • Number of neighbors = 5, reduction by 20% at each iteration, statistically same as best performing feature subset is returned	(Kononenko, 1994; Kira and Rendell, 1992)
L0-norm		(Weston et al., 2003)
Forward Stepwise Selection	using SVM classifier for wrapping	(Caruana and Freitag, 1994)
Koller-Sahami (with backward wrapping by SVM)	• $k = 0$, best performing feature subset is returned • $k = 1$, best performing feature subset is returned • $k = 2$, best performing feature subset is returned • $k = 0$, statistically same as best performing feature subset is returned • $k = 1$, statistically same as best performing feature subset is returned • $k = 2$, statistically same as best performing feature subset is returned	(Koller and Sahami, 1996)
IAMB	• G^2 test and $a = 0.05$ • G^2 test and $a = 0.01$ • mutual information criterion with threshold=0.01	(Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a)
K2MB		(Cooper et al., 1997; Cooper and Herskovits, 1992)

Table 7: Continued from Table 6.

Method	Additional Information	Reference
BLCD-MB		(Mani and Cooper, 2004)
FAST-IAMB	G^2 test and $a = 0.05$	(Yaramakala and Margaritis, 2005)
HITON-PC (semi-interleaved)	• $max\text{-}k = 4$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 2$ and $a = 0.05$ • $max\text{-}k = 1$ and $a = 0.05$ • $max\text{-}k = 4$ and $a = 0.01$ • $max\text{-}k = 3$ and $a = 0.01$ • $max\text{-}k = 2$ and $a = 0.01$ • $max\text{-}k = 1$ and $a = 0.01$ • $max\text{-}k$ and a selected by cross-validation	Novel algorithm
Interleaved HITON-PC	• $max\text{-}k = 4$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 2$ and $a = 0.05$ • $max\text{-}k = 1$ and $a = 0.05$ • $max\text{-}k = 4$ and $a = 0.01$ • $max\text{-}k = 3$ and $a = 0.01$ • $max\text{-}k = 2$ and $a = 0.01$ • $max\text{-}k = 1$ and $a = 0.01$ • $max\text{-}k$ and a selected by cross-validation	(Aliferis et al., 2003a)
MMPC	• $max\text{-}k = 4$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 2$ and $a = 0.05$ • $max\text{-}k = 1$ and $a = 0.05$ • $max\text{-}k = 4$ and $a = 0.01$ • $max\text{-}k = 3$ and $a = 0.01$ • $max\text{-}k = 2$ and $a = 0.01$ • $max\text{-}k = 1$ and $a = 0.01$ • $max\text{-}k$ and a selected by cross-validation	(Tsamardinos et al., 2006, 2003b)
Interleaved MMPC	• $max\text{-}k = 4$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 2$ and $a = 0.05$ • $max\text{-}k = 1$ and $a = 0.05$ • $max\text{-}k = 4$ and $a = 0.01$ • $max\text{-}k = 3$ and $a = 0.01$ • $max\text{-}k = 2$ and $a = 0.01$ • $max\text{-}k = 1$ and $a = 0.01$ • $max\text{-}k$ and a selected by cross-validation	Novel algorithm
HITON-MB (semi-interleaved)	• $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.01$	Novel algorithm
MMMB	• $max\text{-}k = 3$ and $a = 0.05$ • $max\text{-}k = 3$ and $a = 0.01$	(Tsamardinos et al., 2003b)

Table 8: Real data sets used in evaluation of predictivity and compactness. This table is continued in Table 9.

Data set name	Domain	Num. variables	Num. samples	Target	Data type	Cross-val. design	Discretization applied	Notes	Reference
Infant Mortality	Clinical	86	5,337	Died within the first year	Discrete	1-fold cross-val.	Already discrete	Imputed by nearest neighbor method	(Mani and Cooper, 1999)
Ohsumed	Text	14,373	5,000	Relevant to neonatal diseases	Continuous	1-fold cross-val.	Word absent / present		(Joachims, 2002)
ACPJ Etiology	Text	28,228	15,779	Relevant to etiology	Continuous	1-fold cross-val.	Word absent / present		(Aphinyanaphongs et al., 2006)
Lymphoma	Gene expression	7,399	227	3-year survival: dead vs. alive	Continuous	10-fold cross-val.	Binary/ternary univariate; used window sizes 10,15, 20, 25, 30 for ternary		(Rosenwald et al., 2002)
Gisette	Digit recognition	5,000	7,000	Separate 4 from 9	Continuous	1-fold cross-val.	Pixel present / absent	Used original training & validation sets only	NIPS 2003 Feature Selection Challenge (Guyon et al., 2006a)
Dexter	Text	19,999	600	Relevant to corporate acquisitions	Continuous	10-fold cross-val.	Word absent / present	Used original training & validation sets only	NIPS 2003 Feature Selection Challenge (Guyon et al., 2006a)
Sylva	Ecology	216	14,394	Ponderosa pine vs. everything else	Continuous & discrete	1-fold cross-val.	Binary/ternary univariate; used window sizes 1000, 1500, 2000, 2500, 3000 for ternary	Used original training & validation sets only	WCCI 2006 Performance Prediction Challenge

Table 9: Continued from Table 8.

Data set name	Domain	Num. variables	Num. samples	Target	Data type	Cross-val. design	Discretization applied	Notes	Reference
Ovarian Cancer	Proteomics	2,190	216	Cancer vs. normals	Continuous	10-fold cross-val.	Binary/ternary univariate; used window sizes 10, 15, 20, 25, 30 for ternary		(Conrads et al., 2004)
Thrombin	Drug discovery	139,351	2,543	Binding to thrombin	Discrete (binary)	1-fold cross-val.	Already discrete		KDD Cup 2001
Breast Cancer	Gene expression	17,816	286	Estrogen-receptor positive (ER+) vs. ER-	Continuous	10-fold cross-val.	Binary/ternary univariate, used window sizes 10, 15, 20, 25, 30 for ternary		(Wang et al., 2005)
Hiva	Drug discovery	1,617	4,229	Activity to AIDS HIV infection	Discrete (binary)	1-fold cross-val.	Already discrete	Used original training & validation sets only	WCCI 2006 Performance Prediction Challenge
Nova	Text	16,969	1,929	Separate politics from religion topics	Discrete (binary)	1-fold cross-val.	Already discrete	Used original training & validation sets only	WCCI 2006 Performance Prediction Challenge
Bankruptcy	Financial	147	7,063	Personal bankruptcy	Continuous & discrete	1-fold cross-val.	Binary/ternary univariate, used window sizes 1000, 1500, 2000, 2500, 3000 for ternary	Imputed by nearest neighbor method	(Foster and Stine, 2004)

Table 10: Classification performance (AUC) for polynomial SVMs and classifiers native to LARS-EN, L0, and RFVS feature selection algorithms induced with features selected by the latter three methods. In cells marked with "T", the corresponding feature selection method did not terminate within the allotted time.

Feature subset	*Classifier*	Infant_Mortality	Ohsumed	ACPJ_Etiology	Lymphoma	Gisette	Dexter	Sylva	Ovarian_Cancer	Thrombin	Breast_Cancer	Hiva	Nova	Bankruptcy
LARS-EN (w/o stat. comp.)	SVM	0.88	0.80	0.89	0.60	0.99	0.98	1.00	0.98	0.89	0.92	0.73	0.96	0.95
	LARS-EN	0.88	0.81	0.88	0.60	1.00	0.98	1.00	0.99	0.89	0.92	0.77	0.94	0.94
LARS-EN (with stat. comp.)	SVM	0.86	0.77	0.82	0.57	0.99	0.98	1.00	0.96	0.85	0.94	0.62	0.96	0.95
	LARS-EN	0.87	0.78	0.82	0.57	1.00	0.97	0.99	0.96	0.90	0.94	0.69	0.93	0.94
L0	SVM	0.82	0.72	0.84	0.60	0.99	0.97	1.00	0.97	0.81	0.91	0.68	0.96	T
	L0	0.81	0.72	0.87	0.58	0.99	0.97	1.00	0.96	0.81	0.91	0.69	0.95	T
RFVS (w/o stat. comp.)	SVM	0.82	T	T	0.61	T	0.98	1.00	0.97	T	0.93	0.74*	T	0.96
	RF	0.84	T	T	0.63	T	0.98	1.00	0.97	T	0.91	0.78	T	0.97
RFVS (with stat. comp.)	SVM	0.86	T	T	0.61	T	0.98	1.00	0.96	T	0.93	0.68*	T	0.97
	RF	0.78	T	T	0.63	T	0.98	1.00	0.97	T	0.92	0.75	T	0.97

Table 11: Simulated and resimulated data sets used for experiments. *Lung_Cancer* network is resimulated from human lung cancer gene expression data (Bhattacharjee et al., 2001) using SCA algorithm (Friedman et al., 1999b). *Gene* network is resimulated from yeast cell cycle gene expression data (Spellman et al., 1998) using SCA algorithm. More details about data sets are provided in Tsamardinos et al. (2006).

Bayesian network	**Number of variables**	**Training samples**	**Number of selected targets**
Child10	200	5 x 200, 5 x 500, 1 x 5000	10
Insurance10	270	5 x 200, 5 x 500, 1 x 5000	10
Alarm10	370	5 x 200, 5 x 500, 1 x 5000	10
Hailfinder10	560	5 x 200, 5 x 500, 1 x 5000	10
Munin	189	5 x 500, 1 x 5000	6
Pigs	441	5 x 200, 5 x 500, 1 x 5000	10
Link	724	5 x 200, 5 x 500, 1 x 5000	10
Lung_Cancer	800	5 x 200, 5 x 500, 1 x 5000	11
Gene	801	5 x 200, 5 x 500, 1 x 5000	11

Table 12: Algorithms used in local causal discovery experiments with simulated and resimulated data.

HITON-PC (max k=4)	**HITON-PC-FDR** (max k=4)
HITON-PC (max k=3)	**HITON-PC-FDR** (max k=3)
HITON-PC (max k=2)	**HITON-PC-FDR** (max k=2)
HITON-PC (max k=1)	**HITON-PC-FDR** (max k=1)
Interleaved HITON-PC (max k=4)	**HITON-MB** (max k=3)
Interleaved HITON-PC (max k=3)	**MMMB** (max k=3)
Interleaved HITON-PC (max k=2)	**RFE** (reduction of features by 50%)
Interleaved HITON-PC (max k=1)	**RFE** (reduction of features by 20%)
MMPC (max k=4)	**UAF-KruskalWallis-SVM** (50%)
MMPC (max k=3)	**UAF-KruskalWallis-SVM** (20%)
MMPC (max k=2)	**UAF-Signal2Noise-SVM** (50%)
MMPC (max k=1)	**UAF-Signal2Noise-SVM** (20%)
Interleaved MMPC (max k=4)	**L0**
Interleaved MMPC (max k=3)	**LARS-EN** (for multiclass response)
Interleaved MMPC (max k=2)	**LARS-EN** (one-versus-rest)
Interleaved MMPC (max k=1)	

JMLR: Workshop and Conference Proceedings 11:235–284, 2010 Submitted 7/07; Revised 2/08; Published 1/10

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part II: Analysis and Extensions

Constantin F. Aliferis CONSTANTIN.ALIFERIS@NYUMC.ORG
Center of Health Informatics and Bioinformatics
Department of Pathology
New York University
New York, NY 10016, USA

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Center of Health Informatics and Bioinformatics
Department of Medicine
New York University
New York, NY 10016, USA

Ioannis Tsamardinos TSAMARD@ICS.FORTH.GR
Computer Science Department, University of Crete
Institute of Computer Science, Foundation for Research and Technology, Hellas
Heraklion, Crete, GR-714 09, Greece

Subramani Mani SUBRAMANI.MANI@VANDERBILT.EDU
Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA

Xenofon D. Koutsoukos XENOFON.KOUTSOUKOS@VANDERBILT.EDU
Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37212, USA

Editor: Marina Meila

Abstract

In part I of this work we introduced and evaluated the *Generalized Local Learning* (GLL) framework for producing local causal and Markov blanket induction algorithms. In the present second part we analyze the behavior of GLL algorithms and provide extensions to the core methods. Specifically, we investigate the empirical convergence of GLL to the true local neighborhood as a function of sample size. Moreover, we study how predictivity improves with increasing sample size. Then we investigate how sensitive are the algorithms to multiple statistical testing, especially in the presence of many irrelevant features. Next we discuss the role of the algorithm parameters and also show that Markov blanket and causal graph concepts can be used to understand deviations from optimality of state-of-the-art non-causal algorithms. The present paper also introduces the following extensions to the core GLL framework: parallel and distributed versions of GLL algorithms, versions with false discovery rate control, strategies for constructing novel heuristics for specific domains, and divide-and-conquer *local-to-global learning* (LGL) strategies. We test the generality of the LGL approach by deriving a novel LGL-based algorithm that compares favorably to

the state-of-the-art global learning algorithms. In addition, we investigate the use of non-causal feature selection methods to facilitate global learning. Open problems and future research paths related to local and local-to-global causal learning are discussed.
Keywords: local causal discovery, Markov blanket induction, feature selection, classification, causal structure learning, learning of Bayesian networks

1. Introduction

The present paper constitutes the second part of the study of *Generalized Local Learning* (GLL) which provides a unified framework for discovering local causal structure around a target variable of interest using observational data under broad assumptions. GLL supports local discovery of variables that are direct causes or direct effects of the target and of the Markov blanket of the target. In the first part of the work (Aliferis et al., 2010) we introduced GLL and explained the importance of local causal discovery both for identification of highly predictive and parsimonious feature sets (feature selection problem), and for scaling up causal discovery. We then evaluated GLL instantiations against a plethora of state-of-the-art alternatives in many real, simulated and resimulated data sets. The main conclusions were that GLL algorithms achieved excellent predictivity, compactness and ability to learn local neighborhoods. Moreover, state-of-the-art non-causal feature selection methods often achieve excellent predictivity but are misleading in terms of causal discovery.

In the present paper we provide several extensions to GLL, study its properties, and extend to global graph learning using GLL as the core method. Because of the close relationship with Aliferis et al. (2010) we do not repeat here background material, technical definitions, or algorithm specifications. These are found in Aliferis et al. (2010), Sections 2–4.

The paper is organized as follows: Section 2 studies the empirical convergence of GLL instantiations to the true local neighborhood and to optimal predictivity as a function of sample size. Section 3 studies the effects of multiple statistical testing and the sensitivity of GLL algorithms to large numbers of irrelevant features. Section 4 provides a theoretical analysis of GLL algorithms with respect to determinants of statistical decisions, heuristic efficiency and construction of inclusion heuristic functions, reasons for good performance of direct causes and effects instead of induced Markov blanket, and reduced sensitivity to error estimation problems that affect wrappers and traditional filters. Section 5 covers two algorithmic extensions, parallel processing and False Discovery Rate pre-filtering. Section 6 investigates the use of local learners like GLL for global learning and provides a general local-to-global learning framework. In that section we also derive a new algorithm HHC and compare it to the previously described MMHC, and show the potential of local induction variable ordering for tractability and quality improvements. Section 7 uses causal feature selection theory to shed light on limitations of established and newer feature selection methods and the inappropriateness of causally interpreting their output. Section 8 concludes with a discussion of the findings of the present paper and several open problems. An appendix and an online supplement (http://www.nyuinformatics.org/downloads/supplements/JMLR2009/index.html) provide additional results, as well as code and data sets that can be used to replicate the experiments.

2. Empirical Convergence and Comparison of Theoretical to Estimated Markov Blanket

As explained in Aliferis et al. (2010), arguments about the suitability of Markov blanket induction for feature selection for classification are based on large sample results, with convergence of small sample performance to the theoretical optimum being unknown. In the present section we use simulated data sets from published Bayesian networks to produce an empirical evaluation of classification performance convergence with respect to training sample size of two types of classifiers: one that uses the estimated Markov blanket ($MB(T)$) or parents and children set ($PC(T)$) and one that uses the true $MB(T)$ or $PC(T)$ set (obtained from the known generative network). We use polynomial SVMs and KNN to fit each classifier type from three training sample sizes: 200, 500 and 5,000 samples. We note that GLL algorithms provide predictive and optimality guarantees for universal approximator classifiers and SVMs and KNN are used here as exemplars of this class of algorithms. In Aliferis et al. (2010) we also discuss more generally suitable classifiers, distributions and loss functions for GLL instantiations. An independent sample of 5,000 instances is used as evaluation test for classification performance (measured by AUC for binary and proportion of correct classifications for multiclass classification tasks). We use data sets sampled from 9 different Bayesian networks (See Table 15 in the Appendix). For each Bayesian network, we randomly select 10 different targets and generate 5 samples (except for sample size 5,000 where one sample is generated) to reduce variability due to sampling.[1] An independent sample of 5,000 instances is used as evaluation test for classification performance. Several local causal induction algorithms are used (including algorithms that induce direct causes/direct effects, and Markov blankets), and are compared to several non-causal algorithms to obtain reference points for baseline performance: RFE, UAF (univariate association filtering), L0, and LARS-EN (see Table 16 in the Appendix for the list of all algorithms). Classifier parameters (misclassification cost C and degree d for polynomial SVMs and number of neighbors K for KNN) are optimized by nested cross-validation following the same methodology as in Aliferis et al. (2010).

Results are presented in Figure 1 (and more details are given in Tables S19 and S20 of the online supplement). The main conclusions follow. Note that similar patterns are present when KNN is used instead of SVMs (with the only difference that convergence is slightly slower for KNN than for SVMs). For brevity we discuss here the SVM results only.

(a) Classification performance of the true parents and children and Markov blanket feature sets are not statistically significantly different at the 0.05 alpha level in sample 200 (p-value = 0.1440) and are statistically significantly different for larger samples (p-values = 0.0098 and $<$0.0001 for sample sizes 500 and 5,000, respectively). The difference in SVM classification performance between using the $PC(T)$ and $MB(T)$ sets however does not exceed 0.02 AUC in favor of the $MB(T)$ set. This means that even when the true $PC(T)$ and $MB(T)$ sets are known in the tested data, fitting classifiers from small data using the $PC(T)$ set is as good as

1. For networks *Lung_Cancer* and *Gene*, we also add an eleventh target that corresponds to the natural response variable: lung cancer diagnosis and cell cycle state, respectively. For network *Munin* we use only 6 targets because of extreme probability distributions of the majority of variables that do not allow variability in the finite sample of size 500 and even 5000. Because of the same reason, we did not experiment with sample size 200 in the *Munin* network.

using the $MB(T)$ set. In large sample, $MB(T)$ features have a small predictive advantage over $PC(T)$ features.

(b) In small samples, feature selection increases classification performance for all tested classifier types (i.e., both when we know the $PC(T)$ or $MB(T)$ sets and when we estimate them from data) over using all features. This advantage becomes smaller but does not vanish in large sample. The difference in SVM classification performance between an average feature selection method and using all features is

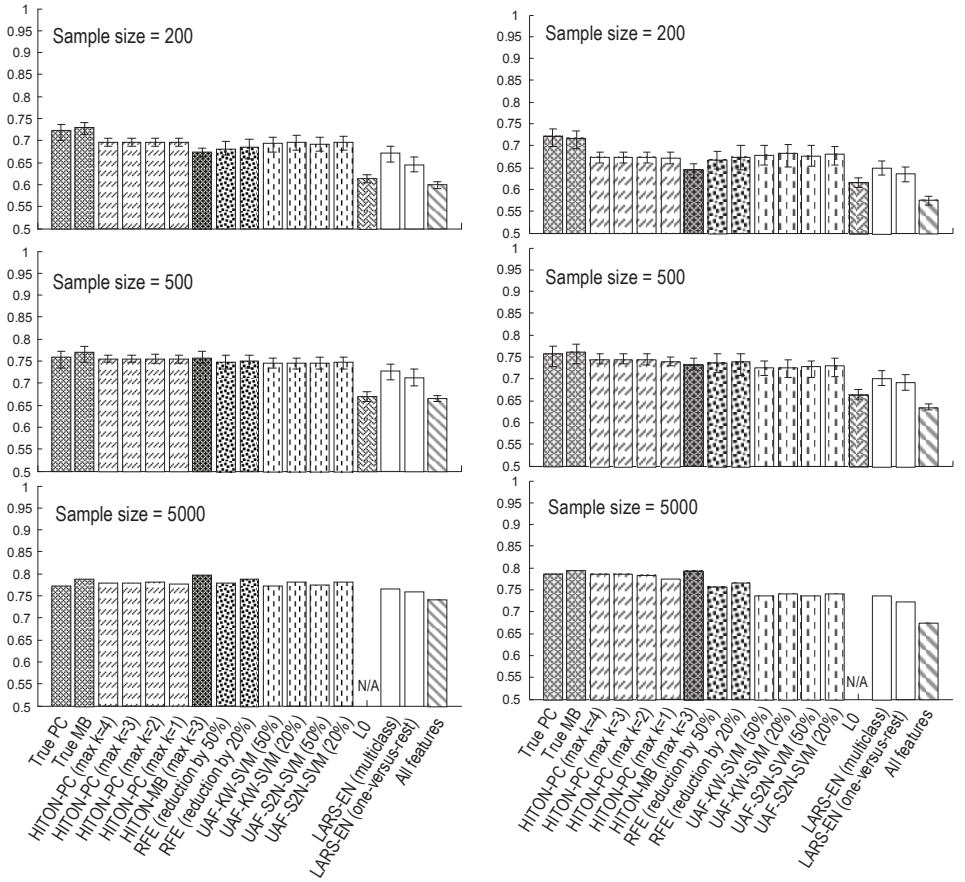


Figure 1: Classification performance of polynomial SVM (left) and KNN (right) classifiers in 9 simulated and resimulated data sets. Results are given for training sample sizes = 200, 500, and 5000. "True-PC" and "True-MB" correspond to the true $PC(T)$ and $MB(T)$ feature sets obtained from the known generative network. The bars denote maximum and minimum performance over multiple training samples of each size (data is available only for sample sizes 200 and 500). The performances reported in the figure are averaged over all data sets, selected targets, and multiple samples of each size. L0 did not terminate within the allotted time limit for sample size 5000.

statistically significant at the 0.05 alpha level (p-values = $<$0.0001, 0.0028, $<$0.0001 for sample sizes 200, 500, and 5,000, respectively).

(c) The true $PC(T)$ or true $MB(T)$ features set when fitted from sample size of 200 has a small (0.02-0.03 AUC/proportion of correct classifications for SVM) advantage over the estimated $PC(T)$ or $MB(T)$ features fitted from small sample. This difference is statistically significant at the 0.05 alpha level with p-values 0.0144 and $<$0.0001 for the $PC(T)$ and $MB(T)$ classifiers, respectively. Very quickly (as sample size becomes 500), this advantage becomes insignificant (0.01 point of AUC/proportion of correct classifications for SVM) with corresponding p-values 0.4708 and 0.0506 for the $PC(T)$ and $MB(T)$ classifiers, respectively. This implies that predictivity of estimated $MB(T)$ and $PC(T)$ sets converge to the optimal one very quickly with respect to sample size.

(d) Classifiers for estimated $MB(T)/PC(T)$ sets fitted from small sample and classifiers for the true $MB(T)/PC(T)$ sets fitted from small sample have indistinguishable performance in sample size 500 (as shown in (c) above); then performance increases in sample size 5,000 for both types of classifiers (p-values ranging from $<$0.0001 to 0.0174 with AUC increases between 0.01 and 0.04). We thus conclude that fitting the right classifier parameters to the identified features is less sample efficient than identifying the right feature set.

(e) Some of the non-causal feature selection methods (e.g., L0, LARS-EN) tend to compare less favorably in small sample to their large sample performance compared to GLL algorithms.

3. Multiple Statistical Tests and Insensitivity to Irrelevant Variables

In this section we focus our attention to a subtle but an important problem facing many feature and causal discovery algorithms operating in very high dimensional spaces, namely the problem of multiple statistical comparisons, which is exacerbated when many irrelevant features are present. We will show that GLL algorithms have inherent control to false positives due to multiple comparisons while the same is not true for other non-causal feature selection methods tested.

Briefly stated, when conducting n statistical tests with an error type I level α (i.e., statistical significance level, that is probability that a truly null hypothesis is rejected, thus falsely concluding that a statistical difference or association or dependence exists when in reality it does not) it is expected that $\alpha \cdot n$ false positives will occur on average. Consider a common analysis situation in bioinformatics research where a researcher conducts one test per variable (i.e., single nucleotide polymorphism (SNP)) in an assay with 10,000 SNP probes in total. 10,000 such tests need be conducted to see whether univariately each SNP probe is differentially present in two or more phenotype categories. If the researcher uses α equal to 5%, then under the null hypothesis (i.e., all 10,000 SNPs are not truly differentially expressed) the analysis will yield 500 false positive SNP probes.

Standard statistical practice involves addressing the problem via one of two basic approaches. The first approach, the classic Bonferroni correction (Casella and Berger, 2002), adjusts the α by replacing it by α/n so that in our example the 5% false positive rate is preserved for each feature selected by the multiple tests. This approach preserves the desired α, but reduces the power to detect statistically significant features (namely

Table 1: Classification performance (AUC) of polynomial SVM estimated on 5,000 sample independent testing set for features selected by HITON-PC with parameter *max-k*={0, 1, 2, 3, 4} on different training sample sizes {100, 200, 500, 1000, 2000, 5000}. The color of each table cell denotes strength of predictivity with yellow (light) corresponding to low classification performance and red (dark) to high classification performance.

Lung_Cancer	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	1.00	0.99	0.99	0.99	0.99	0.97	0.99	0.98	0.98	0.98	0.63	0.63	0.62	0.62	0.62	0.50	0.50	0.50	0.50	0.50
200	1.00	1.00	0.99	0.98	0.98	0.99	1.00	0.99	0.99	0.99	0.67	0.69	0.67	0.66	0.66	0.51	0.50	0.49	0.50	0.50
500	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.72	0.73	0.72	0.71	0.50	0.50	0.51	0.49	0.49
1000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.68	0.74	0.73	0.74	0.72	0.50	0.52	0.51	0.50	0.49
2000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.74	0.74	0.74	0.74	0.49	0.50	0.49	0.50	0.49
5000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72	0.74	0.74	0.74	0.74	0.51	0.51	0.49	0.49	0.49

Alarm10	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	0.95	0.95	0.95	0.95	0.95	0.83	0.92	0.92	0.92	0.92	0.66	0.69	0.69	0.69	0.69	0.50	0.50	0.50	0.50	0.50
200	0.96	0.95	0.95	0.95	0.95	0.89	0.95	0.95	0.95	0.95	0.68	0.77	0.78	0.78	0.78	0.50	0.50	0.50	0.50	0.50
500	0.96	0.96	0.96	0.96	0.96	0.93	0.95	0.95	0.95	0.95	0.71	0.80	0.80	0.80	0.81	0.50	0.51	0.50	0.50	0.50
1000	0.97	0.97	0.97	0.97	0.97	0.94	0.97	0.96	0.96	0.96	0.73	0.82	0.81	0.82	0.82	0.50	0.50	0.50	0.50	0.50
2000	0.97	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.76	0.82	0.82	0.82	0.82	0.50	0.50	0.50	0.50	0.50
5000	0.97	0.98	0.97	0.97	0.97	0.97	0.98	0.97	0.97	0.97	0.81	0.83	0.83	0.83	0.83	0.50	0.50	0.50	0.50	0.50

Low classification performance — *High classification performance*

the features that are truly differentially expressed and detectable at α but non-detectable at α/n), hence creates false negatives that were not present before the correction. The second approach, False Discovery Rate (FDR) control (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995), trades off false positives and false negatives by ensuring not that each feature passing the chosen p-value threshold preserves the original α, but that from the all features found to be significant (i.e., for which the null hypothesis is rejected) a desired proportion will be false positives on average. In our example, FDR methods may, for example, allow the researcher to ensure that on average no more than 10 out of 100 SNPs selected are false positives. This is highly useful in exploratory analysis of high-dimensional data where subsequent experimentation can sort out false positives easily but where false negatives have high cost.

Constraint-based causal methods employ, in large data sets and depending on connectivity and inclusion heuristic efficiency, many thousands of statistical tests of independence and are thus expected a priori to be particularly sensitive to the multiple testing problem. We note that, rather not obviously at first, testing under the null hypothesis does not only occur when irrelevant features exist but also whenever we test weakly relevant features conditioned on a set of variables that blocks all paths connecting it with the target. Other feature selection methods do not explicitly conduct statistical tests of independence but may also be sensitive to many irrelevant features as we will show. In the present section we first systematically explore empirically and then

Table 2: Number of false negatives in the parents and children set for features selected by HITON-PC with parameter *max-k*={0, 1, 2, 3, 4} on different training sample sizes {100, 200, 500, 1000, 2000, 5000}. For Version 4 of the network the parents and children set is empty since there are no relevant variables. The color of each table cell denotes number of false negatives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*				
	max-k parameter														
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	3.30	15.30	18.20	18.20	18.20	3.30	15.40	18.40	18.40	18.40	9.40	21.90	23.40	23.40	23.40
200	1.20	7.70	17.70	19.60	19.60	1.20	7.70	17.70	19.60	19.60	4.40	17.50	23.20	23.40	23.40
500	0.80	1.30	5.70	15.10	18.00	0.80	1.30	5.70	15.10	18.00	1.00	4.60	17.50	21.70	21.90
1000	0.30	1.00	1.50	5.40	11.70	0.30	1.00	1.50	5.40	11.70	0.80	1.70	6.60	17.50	19.90
2000	0.30	0.90	1.00	1.80	4.10	0.30	0.90	1.00	1.80	4.10	0.70	1.00	1.80	8.70	15.80
5000	0.00	0.40	1.00	1.10	1.10	0.00	0.40	1.00	1.10	1.10	0.30	0.80	1.00	1.40	4.80

Alarm10	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*				
	max-k parameter														
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	1.70	4.10	4.10	4.10	4.10	1.70	4.10	4.20	4.20	4.20	2.20	5.00	5.00	5.00	5.00
200	1.40	3.90	4.00	4.00	4.00	1.40	3.90	4.00	4.00	4.00	1.80	4.50	4.70	4.70	4.70
500	0.40	2.60	2.70	2.70	2.70	0.40	2.60	2.90	3.00	3.00	0.60	3.90	4.40	4.40	4.40
1000	0.10	2.00	2.10	2.10	2.10	0.10	2.00	2.20	2.20	2.20	0.80	3.60	3.90	4.00	4.00
2000	0.00	1.40	1.50	1.50	1.50	0.00	1.40	1.50	1.50	1.50	0.10	3.10	3.60	3.50	3.50
5000	0.00	0.50	1.10	1.20	1.20	0.00	0.50	1.10	1.20	1.20	0.00	1.40	1.70	1.80	1.80

Small number of false negatives — *Large number of false negatives*

examine theoretically the degree of sensitivity of GLL algorithms to irrelevant features, how they address the multiple testing problem, and how other feature selection and causal discovery algorithms compare along these dimensions.

In the first set of experiments we run only semi-interleaved HITON-PC without symmetry correction on two networks and variants. The networks, described in Aliferis et al. (2010), are the *Lung_Cancer* resimulated network and the *Alarm10* network. The former is chosen for its higher connectivity whereas the latter is designed to have lower connectivity. In the *Lung_Cancer* network we focused our attention on the natural target variable; this target has 26 members of the parents and children set and 18 spouses, 14 irrelevant variables, and 741 weakly relevant ones. We created four versions of this network: *Version 1* contains the original network (total number of variables 800). In *Version 2* we augment the original network with 7990 irrelevant variables (total number of variables 8790). *Version 3* is the same as Version 2, except for 10% of values of the target are randomly flipped to weaken the signal (total number of variables 8790). Finally, *Version 4* is same as Version 2, except that there are only irrelevant variables and the target (total number of variables is $8790 - 741 - 18 - 26 = 8005$). The tiled *Alarm10* has also four corresponding versions but its target was chosen randomly and it has only 6 members of the parents and children set and no spouses. In both networks (and their

Table 3: Number of false positives (within weakly relevant variables) in the parents and children set for features selected by HITON-PC with parameter *max-k*={0,1,2,3,4} on different training sample sizes {100,200,500,1000,2000,5000}. For Version 4 of the network there are no weakly relevant variables. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*				
	max-k parameter														
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	65.00	0.80	0.30	0.30	0.30	65.00	0.70	0.40	0.40	0.40	62.40	0.90	0.50	0.50	0.50
200	120.50	3.00	0.10	0.00	0.00	120.50	3.00	0.10	0.00	0.00	85.60	2.90	0.60	0.60	0.60
500	149.00	5.80	0.00	0.10	0.00	149.00	5.80	0.00	0.10	0.00	110.70	4.20	0.40	0.30	0.30
1000	202.90	11.60	0.10	0.00	0.00	202.90	11.60	0.10	0.00	0.00	123.70	5.70	0.00	0.00	0.00
2000	236.10	16.40	0.50	0.10	0.00	236.10	16.40	0.50	0.10	0.00	171.10	12.00	0.40	0.00	0.00
5000	410.40	30.80	2.60	0.10	0.00	410.40	30.80	2.60	0.10	0.00	272.60	20.30	1.10	0.00	0.00

Alarm10	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*				
	max-k parameter														
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	22.10	3.70	3.70	3.70	3.70	22.10	2.40	2.40	2.40	2.40	22.50	1.80	1.80	1.80	1.80
200	26.50	0.80	0.80	0.80	0.80	26.50	0.60	0.50	0.50	0.50	25.20	1.30	0.90	0.90	0.90
500	32.20	0.90	0.10	0.10	0.10	32.20	0.80	0.10	0.10	0.10	32.00	1.00	0.20	0.20	0.20
1000	30.20	1.40	0.00	0.00	0.00	30.20	1.30	0.00	0.00	0.00	27.10	0.70	0.10	0.30	0.30
2000	33.50	2.90	0.30	0.30	0.30	33.50	2.80	0.30	0.30	0.30	32.40	1.80	0.60	0.20	0.20
5000	38.00	5.40	0.30	0.20	0.10	38.00	5.30	0.30	0.20	0.10	37.30	3.10	0.20	0.20	0.20

Small number of false positives — *Large number of false positives*

variants) we create irrelevant variables by randomly permuting values of weakly and strongly variables so that the distribution of each variable values is realistic. With these 8 data set versions we can systematically examine the effects of presence of irrelevant variables, strength of predictive signal of features for the target, network connectivity and of the values of the GLL *max-k* parameter (Aliferis et al., 2010).

We run HITON-PC and build SVM classifiers for all networks and variants, varying sample size and the *max-k* parameter, and measure AUC, false negatives, false positives that are weakly relevant, false positives that are irrelevant and total false positives. To ensure that our results are not affected by variability in small samples, we generate 10 random samples of each size and average results.

Tables 1–5 provide evidence for the following conclusions:

(a) Classification performance is mildly or not affected by false positives and false negatives (Table 1). When many false negatives are present, predictivity is compensated by the few remaining strong relevant features plus strongly predictive weakly relevant ones. This implies that classification performance cannot be used to inform us about the presence of false positives/negatives.

Table 4: Number of false positives in the parents and children set for features selected by HITON-PC with parameter *max-k*={0, 1, 2, 3, 4} on different training sample sizes {100, 200, 500, 1000, 2000, 5000}. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	65.20	0.80	0.30	0.30	0.30	476.60	2.30	1.90	1.90	1.90	551.20	12.60	9.10	9.10	9.10	411.60	12.70	9.80	9.80	9.80
200	122.00	3.00	0.10	0.00	0.00	609.10	4.20	0.10	0.00	0.00	557.20	17.80	3.50	3.60	3.60	488.60	17.30	5.80	5.50	5.50
500	149.20	5.80	0.00	0.10	0.00	595.00	7.90	0.00	0.10	0.00	535.60	17.50	1.30	1.50	1.70	446.00	28.10	6.40	5.00	4.90
1000	203.40	11.60	0.10	0.00	0.00	625.60	13.20	0.10	0.00	0.00	536.90	18.40	0.20	0.30	0.30	422.70	31.20	6.90	5.30	5.10
2000	236.90	16.40	0.50	0.10	0.00	645.10	18.00	0.50	0.10	0.00	579.00	23.10	0.80	0.00	0.00	409.00	31.80	6.10	4.00	4.00
5000	411.10	30.80	2.60	0.10	0.00	813.50	32.50	2.60	0.10	0.00	670.40	32.10	1.10	0.00	0.00	403.10	30.90	6.20	4.70	4.10

Alarm10	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	22.10	3.70	3.70	3.70	3.70	414.20	25.40	25.20	25.20	25.20	431.20	28.00	28.20	28.20	28.20	392.10	23.30	23.40	23.40	23.40
200	26.50	0.80	0.80	0.80	0.80	439.40	6.30	4.30	4.30	4.30	453.00	11.60	7.40	7.40	7.40	412.90	19.30	9.70	9.70	9.70
500	32.20	0.90	0.10	0.10	0.10	443.80	4.70	0.90	0.90	0.90	449.90	15.80	4.60	4.10	4.00	411.60	24.40	6.80	6.60	6.60
1000	30.20	1.40	0.00	0.00	0.00	444.30	3.70	0.90	0.60	0.60	427.00	13.30	3.40	3.10	3.00	414.10	22.70	7.20	6.40	6.30
2000	33.50	2.90	0.30	0.30	0.30	415.50	4.40	0.30	0.30	0.30	412.40	11.90	2.40	1.80	1.70	382.00	25.00	8.80	6.50	5.90
5000	38.00	5.40	0.30	0.20	0.10	419.00	6.70	0.40	0.20	0.10	404.40	10.80	1.20	0.50	0.50	381.00	22.90	6.10	5.00	4.90

Small number of false positives — *Large number of false positives*

(b) As expected, false negatives are reduced as sample size grows (because power increases), however they also increase as *max-k* grows, because the number of tests increases as *max-k* grows and thus overall power decreases (Table 2).

(c) When no irrelevant features are present, as sample size grows the number of false positives that are weakly relevant increases if *max-k* is not sufficient to block paths from/to each weakly relevant to/from the target. As *max-k* increases the false positives decrease to the point that they vanish (Table 3). Overall, both false negatives and false positives vanish given enough sample size and sufficient (but not excessive) *max-k*, (i.e., sample size $\geq$ 2,000, *max-k*=2) (Tables 2 and 4).

(d) When irrelevant features are present, as sample size grows the number of false positives that are weakly relevant increases if *max-k* is not sufficient to block paths from/to each weakly relevant to/from the target. As *max-k* increases, the false positives decrease to the point that they vanish (Table 3). False positives due to irrelevant features (Table 5) quickly vanish as *max-k* becomes 2 or higher and this holds as long as sample size is larger than 200. False negatives are not affected by presence of irrelevant features (Table 2). Thus, overall, with enough sample size and right value of *max-k*, both false negatives and false positives vanish (Tables 2 and 4).

(e) When the predictive signal is weaker, both false negatives are increased and false positives within weakly relevant variables are decreased for a given sample size (because power is smaller) (Tables 2 and 3). However false positive irrelevant variables (Table 5) are increased. This is due to the fact that fewer features enter

Table 5: Number of false positives (within irrelevant variables) in the parents and children set for features selected by HITON-PC with parameter *max-k*={0, 1, 2, 3, 4} on different training sample sizes {100, 200, 500, 1000, 2000, 5000}. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	0.20	0.00	0.00	0.00	0.00	411.60	1.60	1.50	1.50	1.50	488.80	11.70	8.60	8.60	8.60	411.60	12.70	9.80	9.80	9.80
200	1.50	0.00	0.00	0.00	0.00	488.60	1.20	0.00	0.00	0.00	471.60	14.90	2.90	3.00	3.00	488.60	17.30	5.80	5.50	5.50
500	0.20	0.00	0.00	0.00	0.00	446.00	2.10	0.00	0.00	0.00	424.90	13.30	0.90	1.20	1.40	446.00	28.10	6.40	5.00	4.90
1000	0.50	0.00	0.00	0.00	0.00	422.70	1.60	0.00	0.00	0.00	413.20	12.70	0.20	0.30	0.30	422.70	31.20	6.90	5.30	5.10
2000	0.80	0.00	0.00	0.00	0.00	409.00	1.60	0.00	0.00	0.00	407.90	11.10	0.40	0.00	0.00	409.00	31.80	6.10	4.00	4.00
5000	0.70	0.00	0.00	0.00	0.00	403.10	1.70	0.00	0.00	0.00	397.80	11.80	0.00	0.00	0.00	403.10	30.90	6.20	4.70	4.10

Alarm10	***Version 1*** *(original network)*					***Version 2*** *(original network + irrelevant variables)*					***Version 3*** *(weakened signal + irrelevant variables)*					***Version 4*** *(only irrelevant variables)*				
	max-k parameter																			
Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100	0.00	0.00	0.00	0.00	0.00	392.10	23.00	22.80	22.80	22.80	408.70	26.20	26.40	26.40	26.40	392.10	23.30	23.40	23.40	23.40
200	0.00	0.00	0.00	0.00	0.00	412.90	5.70	3.80	3.80	3.80	427.80	10.30	6.50	6.50	6.50	412.90	19.30	9.70	9.70	9.70
500	0.00	0.00	0.00	0.00	0.00	411.60	3.90	0.80	0.80	0.80	417.90	14.80	4.40	3.90	3.80	411.60	24.40	6.80	6.60	6.60
1000	0.00	0.00	0.00	0.00	0.00	414.10	2.40	0.90	0.60	0.60	399.90	12.60	3.30	2.80	2.70	414.10	22.70	7.20	6.40	6.30
2000	0.00	0.00	0.00	0.00	0.00	382.00	1.60	0.00	0.00	0.00	380.00	10.10	1.80	1.60	1.50	382.00	25.00	8.80	6.50	5.90
5000	0.00	0.00	0.00	0.00	0.00	381.00	1.40	0.10	0.00	0.00	367.10	7.70	1.00	0.30	0.30	381.00	22.90	6.10	5.00	4.90

Small number of false positives — *Large number of false positives*

the $TPC(T)$ set thus leading to fewer tests that can be performed hence smaller capacity to remove irrelevant false positives. As previously with enough sample and right *max-k*, false positives and negatives are fully eliminated (Tables 2 and 4).

(f) When the data consists only of irrelevant features, false positives (irrelevant) are reduced as *max-k* increases for all sample sizes (Table 5). There is a very small persistent residual number of false positives regardless of how small the sample is or how big the *max-k*. These phenomena happen because the algorithm needs a sufficient number of elements in the $TPC(T)$ set (i.e., tentative parents and children of T) in order to execute conditional independence tests and remove the false positive irrelevant features.

(g) The above trends are remarkably consistent in both networks suggesting that different redundancy and connectivity do not affect the above algorithm behavior.

In the second set of experiments we compare empirically in the above two networks (four variants for each as previously) and 6 sample sizes the following algorithms: semi-interleaved HITON-PC, MMPC, a version of HITON-PC where we pre-filter features by Benjamini FDR control (at FDR rate threshold of 5%) (Benjamini and Yekutieli, 2001), the true $PC(T)$ set extracted from the data generating network (denoted as "True-PC" in Table 6), UAF (univariate association filtering) with Bonferroni correction, UAF with Benjamini FDR control, uncorrected UAF, "wrapped" UAF, RFE, and LARS-EN. Tables 6–9 provide support for the following conclusions:

Table 6: Classification performance (AUC) of polynomial SVM estimated on 5,000 sample independent testing set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with *max-k*=2. The color of each table cell denotes strength of predictivity with yellow (light) corresponding to low classification performance and red (dark) to high classification performance.

Lung_Cancer	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*						***Version 4*** *(only irrelevant variables)*					
	sample size																							
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
True-PC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.71	0.72	0.73	0.72	0.73	0.75	0.50	0.50	0.50	0.50	0.50	0.50
UAF	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	1.00	0.63	0.67	0.67	0.68	0.69	0.72	0.50	0.51	0.50	0.50	0.49	0.51
UAF+Bonferroni	0.97	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00	0.60	0.70	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50
UAF+FDR	0.98	1.00	1.00	1.00	1.00	1.00	0.95	1.00	1.00	1.00	1.00	1.00	0.61	0.72	0.74	0.74	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50
HITON-PC	0.99	0.99	1.00	1.00	1.00	1.00	0.98	0.99	1.00	1.00	1.00	1.00	0.62	0.67	0.73	0.73	0.74	0.74	0.50	0.49	0.51	0.51	0.49	0.49
HITON-PC-FDR	0.97	0.99	1.00	1.00	1.00	1.00	0.94	0.99	1.00	1.00	1.00	1.00	0.61	0.68	0.73	0.73	0.74	0.74	0.49	0.49	0.49	0.49	0.49	0.49
MMPC	0.99	0.99	1.00	1.00	1.00	1.00	0.95	0.99	1.00	1.00	1.00	1.00	0.62	0.67	0.73	0.73	0.74	0.74	0.50	0.49	0.50	0.50	0.50	0.50
LARS-EN	0.91	1.00	1.00	1.00	1.00	1.00	0.97	0.98	1.00	1.00	1.00	1.00	0.64	0.70	0.72	0.71	0.73	0.74	0.49	0.50	0.50	0.50	0.50	0.51
RFE (reduction 50%)	0.97	1.00	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	1.00	0.61	0.65	0.71	0.70	0.73	0.74	0.50	0.50	0.50	0.50	0.49	0.50
RFE (reduction 20%)	0.95	0.99	1.00	1.00	1.00	1.00	0.96	0.98	1.00	1.00	1.00	1.00	0.60	0.68	0.71	0.71	0.74	0.74	0.50	0.50	0.50	0.49	0.50	0.50
UAF-KW-SVM (50%)	0.98	0.99	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	1.00	0.65	0.69	0.73	0.71	0.74	0.74	0.50	0.50	0.50	0.50	0.49	0.50
UAF-KW-SVM (20%)	0.95	0.99	1.00	1.00	1.00	1.00	0.96	0.98	1.00	1.00	1.00	1.00	0.66	0.71	0.73	0.72	0.74	0.74	0.50	0.50	0.50	0.49	0.49	0.51
UAF-S2N-SVM (50%)	0.94	0.99	1.00	1.00	1.00	1.00	0.94	0.99	1.00	1.00	1.00	1.00	0.58	0.64	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50
UAF-S2N-SVM (20%)	0.93	0.98	1.00	1.00	1.00	1.00	0.93	0.98	1.00	1.00	1.00	1.00	0.58	0.67	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.49	0.50	0.50

Alarm10	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*						***Version 4*** *(only irrelevant variables)*					
	sample size																							
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
True-PC	0.97	0.96	0.97	0.97	0.97	0.97	0.97	0.96	0.96	0.97	0.97	0.97	0.83	0.82	0.83	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
UAF	0.95	0.96	0.96	0.97	0.97	0.97	0.83	0.89	0.93	0.94	0.96	0.97	0.66	0.68	0.71	0.73	0.76	0.81	0.50	0.50	0.50	0.50	0.50	0.50
UAF+Bonferroni	0.95	0.96	0.97	0.97	0.98	0.98	0.94	0.96	0.97	0.97	0.98	0.98	0.81	0.82	0.82	0.82	0.83	0.83	0.50	0.50	0.50	0.50	0.50	0.50
UAF+FDR	0.95	0.96	0.97	0.97	0.98	0.98	0.95	0.96	0.97	0.97	0.97	0.98	0.81	0.82	0.82	0.82	0.83	0.83	0.50	0.50	0.50	0.50	0.50	0.50
HITON-PC	0.95	0.95	0.96	0.97	0.97	0.97	0.92	0.95	0.95	0.96	0.97	0.97	0.69	0.78	0.80	0.81	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
HITON-PC-FDR	0.94	0.95	0.95	0.97	0.97	0.97	0.94	0.95	0.96	0.96	0.97	0.97	0.80	0.81	0.81	0.81	0.82	0.82	0.49	0.49	0.49	0.49	0.49	0.49
MMPC	0.95	0.95	0.95	0.97	0.97	0.97	0.91	0.95	0.95	0.97	0.97	0.97	0.67	0.77	0.80	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
LARS-EN	0.94	0.95	0.97	0.97	0.97	0.97	0.93	0.95	0.96	0.96	0.97	0.97	0.78	0.80	0.81	0.81	0.83	0.82	0.50	0.49	0.50	0.50	0.49	0.49
RFE (reduction 50%)	0.94	0.95	0.96	0.97	0.97	0.97	0.93	0.94	0.95	0.96	0.96	0.97	0.72	0.79	0.81	0.82	0.82	0.82	0.50	0.50	0.50	0.49	0.50	0.50
RFE (reduction 20%)	0.94	0.95	0.96	0.96	0.97	0.97	0.91	0.95	0.95	0.96	0.97	0.96	0.73	0.79	0.81	0.81	0.82	0.83	0.49	0.50	0.50	0.50	0.50	0.50
UAF-KW-SVM (50%)	0.94	0.95	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.97	0.75	0.79	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
UAF-KW-SVM (20%)	0.95	0.96	0.96	0.97	0.97	0.98	0.93	0.95	0.96	0.97	0.97	0.97	0.77	0.79	0.82	0.83	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
UAF-S2N-SVM (50%)	0.95	0.96	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.97	0.77	0.80	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.49	0.50	0.50
UAF-S2N-SVM (20%)	0.95	0.96	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.97	0.77	0.78	0.81	0.83	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50

Low classification performance — *High classification performance*

Table 7: Number of false negatives in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with *max-k*=2. For Version 4 of the network the parents and children set is empty since there are no relevant variables. The color of each table cell denotes number of false negatives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*					
	sample size																	
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	3.3	1.2	0.8	0.3	0.3	0.0	3.3	1.2	0.8	0.3	0.3	0.0	9.4	4.4	1.0	0.8	0.7	0.3
UAF+Bonferroni	13.9	6.1	1.5	1.0	0.9	0.2	17.6	8.4	1.8	1.0	1.0	0.5	24.9	19.9	6.7	2.4	1.0	1.0
UAF+FDR	9.2	2.5	0.9	0.5	0.4	0.0	13.4	4.8	1.3	0.9	0.8	0.0	24.0	16.2	3.5	1.3	1.0	0.8
HITON-PC	18.2	17.7	5.7	1.5	1.0	1.0	18.4	17.7	5.7	1.5	1.0	1.0	23.4	23.2	17.5	6.6	1.8	1.0
HITON-PC-FDR	19.3	18.5	5.7	1.5	1.0	1.0	19.2	18.5	5.7	1.5	1.0	1.0	24.7	23.3	17.9	6.6	1.8	1.0
MMPC	18.5	17.7	5.7	1.5	1.0	1.0	18.9	17.7	5.7	1.5	1.0	1.0	23.4	22.8	17.6	6.6	1.8	1.0
LARS-EN	19.9	14.2	8.8	7.9	3.6	1.0	15.9	18.6	10.0	10.0	3.7	1.6	22.8	21.5	18.3	13.4	9.4	10.7
RFE (reduction 50%)	20.7	15.9	9.4	6.1	4.1	1.0	18.8	14.6	13.3	9.2	3.2	1.6	21.1	15.9	7.6	8.6	14.8	12.8
RFE (reduction 20%)	21.9	17.1	10.5	12.5	4.9	2.6	18.7	18.8	11.0	9.1	3.7	2.3	15.6	18.1	8.3	14.3	16.9	12.3
UAF-KW-SVM (50%)	17.5	16.6	5.9	5.3	1.6	0.7	17.8	15.8	8.6	9.8	5.6	1.5	20.1	14.1	10.9	9.3	8.2	7.3
UAF-KW-SVM (20%)	21.0	18.8	10.5	8.3	2.6	0.7	19.1	18.7	10.7	13.2	6.4	1.2	20.5	14.3	12.4	8.1	6.9	7.2
UAF-S2N-SVM (50%)	20.8	17.1	6.0	7.6	2.5	1.3	17.6	16.7	8.4	7.1	7.0	1.9	16.6	15.4	15.6	11.5	8.3	4.9
UAF-S2N-SVM (20%)	23.1	19.9	9.4	10.5	5.1	1.8	20.5	18.5	10.4	11.3	7.0	0.7	19.4	14.8	15.4	12.3	6.6	5.5

Alarm10	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*					
	sample size																	
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	1.7	1.4	0.4	0.1	0.0	0.0	1.7	1.4	0.4	0.1	0.0	0.0	2.2	1.8	0.6	0.8	0.1	0.0
UAF+Bonferroni	4.1	2.7	1.4	1.0	0.5	0.0	4.7	3.2	1.5	1.1	0.7	0.2	5.0	4.4	2.7	1.4	1.0	0.5
UAF+FDR	3.3	2.2	0.8	1.0	0.3	0.0	4.3	2.8	1.4	1.1	0.5	0.0	4.9	3.8	2.4	1.2	0.9	0.2
HITON-PC	4.1	4.0	2.7	2.1	1.5	1.1	4.2	4.0	2.9	2.2	1.5	1.1	5.0	4.7	4.4	3.9	3.6	1.7
HITON-PC-FDR	4.6	4.2	3.2	2.3	1.7	1.0	4.8	4.3	3.2	2.3	1.7	1.0	5.5	4.7	4.4	4.2	3.6	2.1
MMPC	4.1	4.0	3.0	2.4	1.6	1.0	4.3	4.1	3.5	2.4	1.6	1.0	5.0	4.7	4.5	4.2	3.7	2.1
LARS-EN	3.8	3.8	1.7	1.7	1.5	1.4	4.4	4.1	2.5	2.2	1.9	1.4	4.6	4.6	4.6	3.5	2.2	2.0
RFE (reduction 50%)	4.1	3.7	2.1	1.9	2.3	1.5	4.8	4.7	3.2	3.3	2.6	1.8	4.6	4.9	5.2	4.6	4.2	3.6
RFE (reduction 20%)	4.1	3.7	2.4	2.7	2.1	1.8	5.0	4.4	3.4	3.2	2.3	2.0	5.0	5.3	5.0	4.5	3.7	3.3
UAF-KW-SVM (50%)	3.8	3.8	2.2	0.8	0.9	0.4	4.8	3.6	2.4	2.2	1.4	0.1	3.8	4.2	3.4	2.1	2.2	0.8
UAF-KW-SVM (20%)	4.0	3.2	2.4	1.1	0.4	0.0	4.2	3.6	2.4	1.9	1.2	0.0	4.2	4.3	2.7	2.8	1.9	1.2
UAF-S2N-SVM (50%)	3.5	3.6	2.1	1.0	0.8	0.4	4.7	3.8	2.2	2.1	1.5	0.2	5.1	4.4	4.3	3.5	2.7	1.0
UAF-S2N-SVM (20%)	4.3	3.5	2.6	1.3	0.5	0.0	4.9	3.7	2.5	1.9	1.7	0.2	5.0	4.5	3.6	3.0	2.5	1.4

Small number of false negatives — *Large number of false negatives*

(h) Due to strength of signal and redundancy of predictors, AUC reaches the theoretical maximum (provided by the generative network) very quickly and for all methods (Table 6).

(i) When no irrelevant features are present and in the stronger signal setting, simple and FDR-corrected UAF (but not wrapped UAF) has the least false negatives in very small samples (Table 7). As sample size grows all methods reduce their false negatives (Table 7). GLL methods pick up the strongly relevant features without false positives and reach near perfect separation (i.e., 1-2 false negatives and zero false positives) at sample size 1,000 and higher (Table 8). No other method simultaneously minimizes false positives and false negatives as GLL.

(j) In the setting of strong signal with irrelevant features, simple UAF has the least false negatives in very small samples (Table 7) and the largest number of false positives (Table 8).

(k) When the predictive signal is weaker, false negatives are increased and weakly relevant false positives are decreased for a given sample size compared to the stronger signal case (Tables 7 and 8). Simple UAF is again most sensitive in terms of detecting strongly relevant features in smaller samples until sample size 1,000-2,000 where UAF-Bonferroni and UAF-FDR and GLL match the false negative rates (Table 7). As previously, GLL (with HITON-PC and MMPC performing similarly) achieves excellent false positive rates better than those by FDR not only for weakly relevant but also for irrelevant features.

(l) HITON-PC augmented with FDR pre-filtering behaves almost identically as regular HITON-PC except for the case with only irrelevant features in the data where HITON-PC without FDR admits a few false positives (Table 9).

(m) State-of-the-art feature selection methods are prone to select very large numbers of irrelevant features (Table 9).

In conclusion, HITON-PC and by extension GLL algorithms (since the same fundamental mechanisms for variable inclusion and elimination are shared because of the GLL-PC template and admissibility requirements), have a very strong built-in capacity to control for false positives due to multiple comparisons. False positives due to multiple comparisons quickly vanish for *max-k* 1 or higher *regardless of sample size*. Given enough sample size (∼1,000 or more in the data tested), and by choosing 5% as the nominal α for all conditioning independence tests executed, the algorithm fully eliminates irrelevant features from its output without incurring a penalty in false negatives, even when irrelevant features are the majority among observed features. Parameter *max-k* controls the false positives due to both weakly relevant and irrelevant features. The false positive rate in this worst-case situation is in the presented experiments $\sim 5/8,000 = 0.000625$ which is much better than what the conservative Bonferroni-adjusted α guarantees, and *without* incurring false negatives (as both Bonferroni and FDR methods do). Both established feature selectors such as variants of UAF and newer ones are very sensitive to irrelevant features and produce large numbers of false positives. Given the attractive characteristics of FDR-augmented HITON-PC, we evaluate it with real data sets in Section 5.

4. Theoretical Analysis of GLL

In the present section we provide a theoretical analysis of the Generalized Local Learning algorithms.

4.1. Determinants of Quality of Statistical Decisions and Computational Tractability. Parameters *max-k* and *h-ps*

On a rather superficial level when conditioning sets are large enough, statistical tests become less reliable. For example, as explained in Aliferis et al. (2010), cells in contingency tables used to calculate p-values of discrete tests of independence (such as the widely-used G^2 or X^2 test) become scarcely populated and this leads to unreliable test results. This motivates the heuristic practice of considering as unreliable and not

Table 8: Number of false positives (within weakly relevant variables) in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with *max-k*=2. For Version 4 of the network there are no weakly relevant variables. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*					
	sample size																	
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	65.0	120.5	149.0	202.9	236.1	410.4	65.0	120.5	149.0	202.9	236.1	410.4	62.4	85.6	110.7	123.7	171.1	272.6
UAF+Bonferroni	1.8	8.9	33.6	65.5	91.6	160.3	0.6	4.1	21.2	52.5	80.3	134.3	0.1	0.7	4.8	14.9	43.4	83.6
UAF+FDR	9.4	39.3	78.3	130.5	168.6	359.9	2.7	13.6	46.2	82.6	111.8	230.7	0.1	2.3	13.3	33.5	70.8	123.6
HITON-PC	0.3	0.1	0.0	0.1	0.5	2.6	0.4	0.1	0.0	0.1	0.5	2.6	0.5	0.6	0.4	0.0	0.4	1.1
HITON-PC-FDR	0.2	0.0	0.0	0.1	0.3	1.4	0.1	0.1	0.0	0.1	0.3	1.4	0.1	0.6	0.3	0.0	0.3	0.5
MMPC	0.3	0.1	0.0	0.1	0.5	2.7	0.3	0.1	0.0	0.1	0.5	2.7	0.7	0.8	0.4	0.0	0.4	1.1
LARS-EN	7.5	15.7	5.7	3.7	39.2	59.0	4.6	2.1	4.9	1.1	4.0	25.7	5.4	2.9	3.4	4.4	7.2	3.2
RFE (reduction 50%)	0.7	7.1	13.1	22.0	79.1	123.2	3.1	5.5	1.7	5.8	20.3	24.1	82.9	43.5	170.5	108.2	152.6	96.8
RFE (reduction 20%)	0.4	3.2	12.1	3.0	73.1	167.9	4.8	1.3	5.5	1.9	14.0	22.2	141.5	28.1	115.1	18.8	122.6	112.9
UAF-KW-SVM (50%)	2.0	1.5	76.5	6.8	124.9	172.8	1.7	3.3	14.9	2.6	37.7	120.2	8.8	83.0	24.1	257.0	83.5	97.3
UAF-KW-SVM (20%)	0.6	1.1	4.8	2.5	91.4	179.9	1.0	2.1	14.1	0.7	10.3	124.4	6.4	82.5	22.4	137.8	19.1	46.9
UAF-S2N-SVM (50%)	1.3	1.4	43.1	2.7	114.3	139.8	3.5	2.1	7.1	5.0	26.9	109.5	228.9	98.4	25.4	102.6	86.6	180.0
UAF-S2N-SVM (20%)	0.2	0.4	12.7	1.2	70.1	128.1	1.0	1.5	5.3	1.6	22.3	120.8	153.4	117.5	19.5	53.8	93.1	175.8

Alarm10	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*					
	sample size																	
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	22.1	26.5	32.2	30.2	33.5	38.0	22.1	26.5	32.2	30.2	33.5	38.0	22.5	25.2	32.0	27.1	32.4	37.3
UAF+Bonferroni	4.4	4.8	7.4	8.6	10.7	14.6	3.3	4.4	6.0	8.0	9.2	13.1	1.5	3.1	4.9	6.7	7.7	10.3
UAF+FDR	5.0	6.2	9.7	10.1	14.3	20.1	3.9	4.8	7.2	8.6	10.7	14.6	1.8	3.8	5.4	7.3	8.7	12.2
HITON-PC	3.7	0.8	0.1	0.0	0.3	0.3	2.4	0.5	0.1	0.0	0.3	0.3	1.8	0.9	0.2	0.1	0.6	0.2
HITON-PC-FDR	0.9	0.5	0.0	0.1	0.1	0.0	0.7	0.4	0.1	0.1	0.1	0.0	0.7	0.6	0.2	0.2	0.2	0.3
MMPC	3.7	0.8	0.2	0.3	0.4	0.1	2.6	0.5	0.2	0.2	0.4	0.1	2.6	0.7	0.3	0.4	0.5	0.3
LARS-EN	20.7	9.4	56.1	24.7	17.2	36.7	3.2	3.0	3.9	4.1	3.9	9.1	1.0	1.6	2.3	3.3	3.4	4.9
RFE (reduction 50%)	16.7	18.6	114.9	68.9	23.7	36.9	2.0	1.3	3.5	2.9	1.5	3.7	19.7	1.4	1.3	1.6	1.9	2.9
RFE (reduction 20%)	11.3	18.1	56.0	9.8	19.7	38.7	2.5	0.9	1.9	2.5	1.7	3.3	11.6	0.9	0.8	1.1	1.5	2.7
UAF-KW-SVM (50%)	13.5	4.0	32.6	51.4	49.7	35.9	3.4	3.4	5.6	5.4	9.1	15.4	13.7	3.7	4.4	5.7	7.6	10.6
UAF-KW-SVM (20%)	5.7	5.4	10.2	42.3	37.5	58.7	3.3	3.1	5.4	5.7	8.8	14.7	5.6	3.3	4.9	5.2	7.3	9.0
UAF-S2N-SVM (50%)	18.6	4.3	72.3	55.0	37.5	38.2	2.0	3.3	8.1	5.9	8.9	14.6	1.4	2.3	2.7	4.2	6.0	9.8
UAF-S2N-SVM (20%)	7.1	4.1	44.6	17.8	38.2	40.1	1.9	3.8	5.0	6.1	8.1	13.1	1.4	2.8	3.2	4.6	6.5	8.8

Small number of false positives — *Large number of false positives*

executing a test in which the sample size is less than: ("number of cells to be fitted" · *h-ps*), with parameter *h-ps* set to 10 by default in the PC algorithm (Spirtes et al., 2000) and 5 in GLL instantiations. Recall from Aliferis et al. (2010) that *h-ps* stands for "heuristic power size" and denotes the smallest sample size per cell in the contingency table of a reliable conditional test of independence. Moreover, when the conditioning set size is large enough to block all paths between a weekly relevant variable and the target, there is no need to exceed this conditioning set size because the resulting tests are redundant and the operation of the algorithm becomes unnecessarily slow. Thus it seems reasonable that we would wish to restrict the conditioning set size to not exceed this sufficient blocking size. This is accomplished by setting the value of parameter *max-k*. We will see however that *max-k* has a much more elaborate function than simply "trimming away" excessive computations.

In reality things are significantly more complicated because, as first pointed out by Spirtes et al. (2000), statistical reliability of a single test is a misleading concept in the context of complex constraint-based algorithms such as GLL. Standard statistical

Table 9: Number of false positives(within irrelevant variables) in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with *max-k*=2. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

Lung_Cancer	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*						***Version 4*** *(only irrelevant variables)*					
	sample size																							
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	0.2	1.5	0.2	0.5	0.8	0.7	411.6	488.6	446.0	422.7	409.0	403.1	488.8	471.6	424.9	413.2	407.9	397.8	411.6	488.6	446.0	422.7	409.0	403.1
UAF+Bonferroni	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0
UAF+FDR	0.0	0.1	0.1	0.0	0.1	0.2	0.5	1.1	3.7	4.8	6.7	12.5	0.2	0.7	1.4	3.2	4.8	8.0	0.1	0.0	0.0	0.1	0.1	0.1
HITON-PC	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	8.6	2.9	0.9	0.2	0.4	0.0	9.8	5.8	6.4	6.9	6.1	6.2
HITON-PC-FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.3	0.1	0.2	0.0	0.1	0.0	0.0	0.1	0.1	0.1
MMPC	0.0	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	8.7	3.4	0.8	0.2	0.4	0.0	8.8	6.4	7.2	6.3	6.7	6.3
LARS-EN	0.4	0.1	0.1	0.0	0.9	0.5	32.8	11.5	29.9	1.4	6.2	52.2	53.2	20.8	32.9	47.3	82.0	39.2	35.9	33.5	69.3	63.0	69.6	38.7
RFE (reduction 50%)	0.0	0.0	0.2	0.5	1.4	1.8	24.8	20.5	4.4	10.7	68.5	78.4	868.9	449.1	1741.5	1132.5	1600.0	1005.6	462.4	1084.1	971.3	918.2	1844.9	223.7
RFE (reduction 20%)	0.0	0.0	0.2	0.0	1.4	3.2	28.3	3.5	21.0	4.9	49.3	78.1	1548.4	252.6	1145.1	183.0	1282.2	1171.0	531.2	106.0	1488.3	103.8	849.0	112.7
UAF-KW-SVM (50%)	0.0	0.0	0.7	0.0	0.5	1.6	1.8	0.2	13.9	0.0	23.5	95.5	56.3	798.3	111.0	2593.7	800.6	801.1	809.3	319.6	1161.8	193.0	1676.1	886.0
UAF-KW-SVM (20%)	0.0	0.0	0.0	0.0	0.5	0.4	1.3	0.1	11.9	0.0	0.0	76.9	47.4	816.0	108.9	1215.2	3.4	12.4	971.0	346.6	1061.1	72.3	1283.0	870.6
UAF-S2N-SVM (50%)	0.0	0.0	0.5	0.0	0.7	0.2	29.8	5.8	3.5	0.0	5.6	49.9	2420.2	911.2	193.5	993.5	803.4	1604.8	676.2	1414.0	1666.8	2540.1	2491.3	1032.9
UAF-S2N-SVM (20%)	0.0	0.0	0.1	0.0	0.3	0.3	7.7	3.3	2.8	0.0	16.5	95.9	1624.5	1077.3	128.1	416.3	805.6	1608.2	1036.1	537.2	819.0	236.2	1279.7	990.0

Alarm10	***Version 1*** *(original network)*						***Version 2*** *(original network + irrelevant variables)*						***Version 3*** *(weakened signal + irrelevant variables)*						***Version 4*** *(only irrelevant variables)*					
	sample size																							
FS method	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	0.0	0.0	0.0	0.0	0.0	0.0	392.1	412.9	411.6	414.1	382.0	381.0	408.7	427.8	417.9	399.9	380.0	367.1	392.1	412.9	411.6	414.1	382.0	381.0
UAF+Bonferroni	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0
UAF+FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.7	0.8	1.0	1.2	0.4	0.2	1.0	1.0	0.6	1.4	0.0	0.0	0.2	0.0	0.1	0.0
HITON-PC	0.0	0.0	0.0	0.0	0.0	0.0	22.8	3.8	0.8	0.9	0.0	0.1	26.4	6.5	4.4	3.3	1.8	1.0	23.4	9.7	6.8	7.2	8.8	6.1
HITON-PC-FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.3	0.2	0.5	0.2	0.1	0.0	0.0	0.0	0.2	0.0	0.1	0.0
MMPC	0.0	0.0	0.0	0.0	0.0	0.0	29.0	3.4	1.2	1.0	0.0	0.1	37.2	5.9	4.4	3.2	2.8	1.0	26.8	10.4	8.5	7.8	9.3	5.8
LARS-EN	0.0	0.0	0.0	0.0	0.0	0.0	34.2	31.7	32.0	21.3	7.7	54.6	16.3	12.9	19.2	45.2	44.6	38.3	55.6	23.8	32.9	9.8	31.9	73.5
RFE (reduction 50%)	0.0	0.0	0.0	0.0	0.0	0.0	12.2	2.9	32.0	18.7	6.4	25.3	385.9	13.1	5.1	7.3	16.2	33.0	502.6	1819.2	80.6	1940.8	965.3	1090.3
RFE (reduction 20%)	0.0	0.0	0.0	0.0	0.0	0.0	11.7	2.6	8.3	23.3	7.2	20.8	201.2	7.4	2.2	8.5	16.7	30.8	264.8	732.4	735.0	633.0	790.6	237.2
UAF-KW-SVM (50%)	0.0	0.0	0.0	0.0	0.0	0.0	16.2	4.0	1.9	2.0	16.0	39.3	240.9	34.1	3.0	8.6	11.8	29.7	204.9	142.9	447.9	2604.5	811.1	1504.0
UAF-KW-SVM (20%)	0.0	0.0	0.0	0.0	0.0	0.0	15.3	2.1	0.9	5.5	13.6	35.8	74.0	23.4	3.0	3.2	8.8	11.3	219.5	102.5	124.6	1554.9	784.1	1230.1
UAF-S2N-SVM (50%)	0.0	0.0	0.0	0.0	0.0	0.0	3.1	5.2	30.2	2.5	20.0	37.2	9.6	2.7	1.1	2.9	3.3	26.4	615.1	32.7	265.8	1901.0	759.3	721.5
UAF-S2N-SVM (20%)	0.0	0.0	0.0	0.0	0.0	0.0	3.1	7.2	3.9	2.4	14.5	23.1	12.3	17.1	2.9	3.7	4.1	10.1	291.8	772.0	392.7	701.5	120.9	870.6

Small number of false positives — *Large number of false positives*

considerations of the type of testing a hypothesis once do not carry over well to the constraint-based algorithm setting. Similarly, running time is also a complex function of direct or indirect restrictions placed on number of tests and the number of variables with which to build such tests (i.e., the size of $TPC(T)$).

We first explain what happens when running semi-interleaved HITON-PC in faithful distributions (same arguments can be generalized to other GLL-PC and GLL-MB versions). Consider first that in the case of a strongly relevant feature S, when conducting just one test $\mathrm{I}(S, T \mid \emptyset)$ for the purposes of inclusion of S in $TPC(T)$, regardless of how small power is, we should always execute this test because the worst that can happen is that we fail to include S in $TPC(T)$, whereas if we do not execute the test and assume independence by default, we will surely miss it. In the context of *many tests* however, the notion of single-test reliability for S no longer applies. For example, when we consider a test that has the potential to reject S from $TPC(T)$ (where it was placed previously by a *different* test), by allowing the conditioning test size to grow large, the power is reduced (assuming monotonic association of S through the potentially multiple paths connecting S with T). Hence, we need to preserve the combined power (i.e., combination of individual powers of all tests applied to S) in order to not eliminate S from $TPC(T)$. Although these tests are highly correlated and combined power is larger than the product of powers of the same set of tests performed on independent samples, still the more tests are executed the smaller the combined power and the larger the possibility of falsely eliminating S becomes. The parameter *h-ps* partially controls power because the larger it is, the smaller number of tests (that would eliminate S) are executed. However *h-ps* should not be too large either because a strongly relevant S will not be included in $TPC(T)$ in the first place. Parameter *max-k* also controls in part the number of tests allowed. *Max-k* does not fully determine the number of tests because it specifies the dimensionality of allowed tests, not their total number. As *max-k* grows, more tests for eliminating S from $TPC(T)$ are executed, thus the combined power drops. In summary, for a given distribution the number of tests performed is affected by *h-ps*, *max-k* and the size of $TPC(T)$.

So far the discussion has centered on one type of conditional independence test, that is, tests where the candidate member of $PC(T)$, X, is a strongly relevant feature (type 1). This is the first of four types of conditional tests. The other three are: conditional independence tests where the candidate member of $PC(T)$, X, is a weakly relevant feature and some paths with T are not blocked by the conditioning set (type 2a), conditional independence tests where the candidate member of $PC(T)$, X, is a weakly relevant feature and all paths with T are blocked by the conditioning set (type 2b), and finally conditional independence tests where the candidate member of $PC(T)$, X, is an irrelevant feature (type 3).

The quality of conditional tests of the first type is determined by the *power* of the association of X with T given the conditioning set. Since not one but potentially many such tests are conducted, the combined power of all such tests determines whether X will be selected and stay in the $TPC(T)$ set. For example, variable X (a true member of $PC(T)$) will be considered for inclusion in $TPC(T)$ by HITON-PC with probability = power of detecting $\neg\mathrm{I}(X, T)$ given the available sample size and test employed. However for X to stay in $TPC(T)$ until the algorithm terminates, and assuming B, C have entered $TPC(T)$, none of the tests $\mathrm{I}(X, T \mid B)$, $\mathrm{I}(X, T \mid C)$, $\mathrm{I}(X, T \mid \{B, C\})$ must conclude independence. The power or each one of these tests can be lower or higher than the power of $\mathrm{I}(X, T)$ and the combined power can quickly diminish, however several mitigating factors prevent this from happening. First, when using linear tests

under common distributional assumptions such as multivariate normality, the necessary sample size to achieve desired level of power grows linearly to number of variables in the conditional set. Second, as explained earlier, conditional independence tests of the same variable and T in the same sample are highly correlated. Third, controlling the number of members of $TPC(T)$ by a good heuristic inclusion function reduces the total number of tests; such control occurs indirectly by putting first the true members of $PC(T)$ or members that block many variables. Fourth, the order of executing the tests and constructing conditioning sets is important for reducing the number of tests performed on strongly relevant variables. This is exemplified in semi-interleaved HITON-PC where new entrants in $TPC(T)$ are tested before current $TPC(T)$ members thus if the heuristic inclusion function is a good one, strongly relevant members are tested a smaller number of times at the elimination phase.

Returning our attention to the quality of statistical decisions for weakly relevant variables, we observe that when a conditioning set *does not* block all paths to/from T either for inclusion or for elimination purposes (type 2a), we are sampling under the alternative hypothesis (i.e., there exists association) and the determining factor for failing to reject the weakly relevant feature is the combined power which is determined by the same factors as elaborated for strongly relevant variables previously. The combined probability for rejection may be small for similar reasons as type 1 conditional independence tests (albeit higher than for strongly relevant features due to the fact that under a good inclusion heuristic weakly relevant features enter $TPC(T)$ later than strongly relevant ones and thus more tests are applied on each weakly relevant than on each strongly relevant feature on average).

However, when the conditioning set blocks all paths from/to T (type 2b), *then we sample under the null hypothesis* and the determining factor shifts from the combined power to the *combined* α (i.e., statistical significance). Given that the α for each conditional test is typically low (i.e., 5% or smaller) and that as the number of tests under the null increases, the combined α drops up to exponentially fast, and eliminating weakly relevant features occurs with high probability as the number of applied tests increases. In HITON-PC, the smaller is *h-ps*, the easier it is to include a weakly relevant feature (based on univariate association heuristic), whereas *max-k* does not affect this function. In terms of rejecting a weakly relevant feature in $TPC(T)$, the larger *max-k* and the smaller *h-ps* become, the easier it is to eliminate a weakly relevant feature.

The quality of statistical decisions for type 3 of conditional independence tests, that is for irrelevant variables, is determined by the combined α since we *always* test under the null hypothesis. Because the combined α drops fast as the number of tests applied to each irrelevant variable (and these tests are abundant when even a handful of variables have been admitted in $TPC(T)$), the combined probability for admitting and not rejecting irrelevant variables is exceedingly small. However when no strongly (and thus no weakly) relevant feature exists, conditioning sets inside the $TPC(T)$ set become smaller as irrelevant variables are eliminated from it with the end result of leaving a small number of "residual" irrelevant features in the final output as evidenced in the simulation experiments of Section 3. By pre-filtering variables with an FDR filter (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995), we not only gain the security that if the data consists exclusively of irrelevant variables fewer or no false positives will be returned, but also we can use *max-k* to control sensitivity and specificity trading weakly relevant false positives for strongly relevant true positives and vice versa (i.e., without worrying about adversely trading off irrelevant features).

		Number of conditional independence tests			*Cost of conditional independence tests*			*Number of false positives (fp) and false negatives (fn)**	
Lung_Cancer	max-k	**HITON-PC**	**MMPC**	max-k	**HITON-PC**	**MMPC**	max-k	# of fn	# of fp
	1	4,028	5,683	1	7,257	8,900	1	1	13
Target variable #1	2	12,328	14,577	2	33,018	38,892	2	1	0
Number of members in	3	73,554	77,885	3	277,922	294,211	3	1	0
PC set = 26	4	250,560	259,099	4	1,181,889	1,225,682	4	3	0
Alarm10	max-k	**HITON-PC**	**MMPC**	max-k	**HITON-PC**	**MMPC**	max-k	# of fn	# of fp
	1	457	490	1	545	585	1	1	2
Target variable #199	2	470	496	2	608	652	2	1	0
Number of members in	3	491	521	3	692	752	3	1	0
PC set = 6	4	496	527	4	717	782	4	1	0

* Results are same for HITON-PC and MMPC for number of false positives and false negatives

Figure 2: Efficiency of HITON-PC versus MMPC.

Finally, the total number of tests is determined by both parameters *h-ps* and *max-k*, in a non-monotonic manner. That is, whenever *h-ps* is extremely large it effectively disallows most tests and the algorithm quickly terminates returning the empty set regardless of *max-k*. For medium/small values of *h-ps*, more tests are executed, more variables enter $TPC(T)$, and many tests are executed before $TPC(T)$ is finalized. *Max-k* modifies this number by potentially restricting the number of tests. When *h-ps* is very small, tests are allowed with very large conditioning tests and as long as *max-k* does not disallow them, the total number of tests grow very large.

4.2. Efficiency and Heuristic Robustness of HITON-PC Versus MMPC

Figure 2 presents the number and cost[2] (proportional to time) of conditional independence tests performed by semi-interleaved HITON-PC versus MMPC in the 2,000-sample data set from the *Alarm10* and *Lung_Cancer* networks. As can be seen, HITON-PC performs fewer tests on average while achieving the same performance as MMPC. We notice that the max-min association heuristic closely reflects the logic behind the combined probability for error for the weakly relevant features. MMPC when testing under the alternative hypothesis (i.e., strongly relevant features, or unblocked weakly relevant ones) requires measuring all relevant associations, whereas HITON requires just the univariate ones *for inclusion purposes*. However semi-interleaved HITON tries to eliminate the newly included variable immediately upon inclusion and thus effectively conducts a similar number of tests as MMPC. Both algorithms when testing under the null hypothesis (irrelevant or fully-blocked weakly relevant features) on average execute the same number of tests. The max-min association inclusion heuristic is a priori more prone to basing its decisions for inclusion in $TPC(T)$ on less statistically reliable criteria. This is because the more associations are considered and the larger the conditioning sets are, the higher variance in the minimum association estimates is expected, making the maximum of such associations over all variables considered more prone to sampling error (i.e., it is likely to be overfitted to the sample). Because of better robustness of

2. The cost of a conditional independence test is calculated as the number of variables participating in it (excluding target variable). For example, univariate tests have cost = 1, tests with conditioning on two variables have cost = 3.

the univariate association relative to the weakest association over many conditional associations true members of $PC(T)$ may enter the $TPC(T)$ set earlier. However both HITON-PC and MMPC exhibit similar performance in real and simulated data sets, demonstrating that the theoretical problem with max-min association is in practice very rare.

4.3. Synthesis and Problems for Inclusion Heuristics; Constructing New Inclusion Heuristics

A problem when inducing local neighborhoods and particularly Markov blankets is that of *information synthesis*. The problem consists of a variable X that is not in $PC(T)$ having higher association (univariate or conditional on some subsets) with T than members of $PC(T)$ (for a concrete example see Figure 13). We will call such variables, *synthesis variables*. Synthesis variables were identified as major problems for algorithms such as IAMB (Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a) or GS (Margaritis and Thrun, 1999) that induce Markov blankets and do so by conditioning in their inclusion phase on all variables in the tentative $MB(T)$. Because of the requirement to condition on all variables in the tentative $MB(T)$, the sample requirements grow exponentially fast to the size of the tentative $MB(T)$ and thus it is absolutely imperative to keep out of it synthesis variables since they unnecessarily increase the sample requirements to the point that the algorithm may need to stop executing conditional independence tests (and either halt or output the tentative $MB(T)$ as best but flawed estimate of the true $MB(T)$).

With regards to GLL algorithms, most efficient operation is achieved when the variables that alone or in combination have the property that block the largest fraction of weakly relevant variables, enter first in $TPC(T)$ (even if they are not strongly relevant themselves). Synthesis variables may or may not have this property, so synthesis may or may not be a problem for a specific GLL algorithm based on characteristics of the specific data in hand.

Construction of new inclusion heuristics may be required in difficult cases where the univariate and max-min heuristics do not work well leading to very slow processing time and very large $TPC(T)$ sets, in order to make operation of local learning tractable. In practice, both the univariate and max-min association heuristics work very well with real and simulated data sets, so we do not pursue here implementation and testing new heuristics in artificial problems, although we recognize the possibility of such need in future problematic data distributions. We outline here, in broad strokes, general strategies for creating new inclusion heuristics for such cases:

1. *Random heuristic search informed by standard heuristic values.* This strategy is based on using one of the usual heuristics to rank candidate variables and making selection decisions based on random selection of a candidate variable with probability proportional to the original heuristic value. This enables using the older heuristic as a starting point but allowing occasionally deviations from it to explore the possibility that lower-ranked candidates may have better potential as blocking variables. A simulated-annealing determination of probability of selection (or other efficient stochastic search algorithms) can be pursued as well.

2. *Constructing new heuristic functions by observing blocking capability* (in terms of candidate variables blocked by conditioning sets in which V is a member) or *probability of a variable V to remain in $TPC(T)$*. The empirical observations can

be collected from a variety of tractable sources: either from a single incomplete run of the algorithm (i.e., without waiting to terminate), or in other data sets characteristic of the domain, or in multiple runs on smaller (randomly chosen) subsets of the original feature set. The new heuristic function F can be constructed as the conditional probability:

$$F(V_i) = P(V_i \in TPC(T) \mid h(V_i))$$

where $h(V_i)$ is the original heuristic value of variable V_i, or the proportion of candidates blocked by a conditioning set containing V_i:

$$F(V_i) = \sum_{k=1}^{M} N_k(V_i)/M$$

where $N_k(V_i)$ is the number of candidate variables blocked by a conditioning set that contains variable V_i in trial k.

3. *Exploiting known domain structure.* When properties of the causal structure of the data generating structure and/or distributional characteristics are known, one can use this information alone or in conjunction with the previous two strategies to derive more efficient heuristics.

We note that developing an inclusion heuristic that leads to efficient execution of GLL is not always feasible since the very problem of finding the features with direct edges with the target is intractable in the worst case (e.g., consider a graph that is fully connected). In some cases, as we will show in Section 6, *it is possible to transform an intractable local learning problem into a tractable one by employing a global learning strategy (i.e., exploiting asymmetries in connectivity).*

4.4. Inductive Bias of GLL

Informally the inductive bias of GLL is that it seeks a balance of false negatives for strongly relevant variables with false positives for weakly relevant and irrelevant variables. The main regulating parameters (for standard inclusion heuristics, elimination and interleaving strategies) are *h-ps* and *max-k*. In practice, the algorithms tested in our work to date reveal higher sensitivity to *max-k* and thus at first approximation we treat optimization of this parameter as having higher priority. Smaller *max-k* empirically decreases false negatives and increases false positives overall. Larger *max-k* increases the false negatives and decreases the false positives. GLL in moderate to large samples achieves small numbers of false negatives and small numbers of false positives. In very small samples GLL prefers false positive errors than false negative ones when *max-k* is small. This occurs because given *some* evidence in favor of $PC(T)$ membership (provided by lower-dimensional and thus more sample efficient) tests of a variable X but *no reliable proof* to the contrary (provided by omitted higher-dimensional and thus unreliable tests), the algorithm outputs X as member of $PC(T)$. A similar behavior exists for the $MB(T)$ versions (with respect to $MB(T)$ membership). Notice that as *max-k* grows many more tests can be executed provided that a liberal *h-ps* is chosen, and these tests can be used to eliminate both weakly relevant as well as strongly relevant features in $TPC(T)$. The choice of a more liberal *h-ps* default value in GLL (compared to the more stringent value in the published implementation of PC algorithm) allows

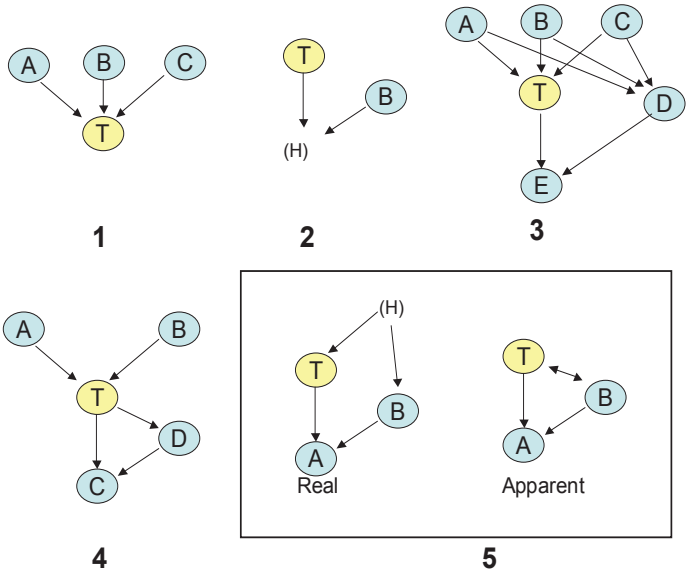


Figure 3: Scenarios explaining good empirical performance of $PC(T)$ set for classification.

a more effective control of the tradeoff between false positives and false negatives in small samples by changing values of *max-k*.

By contrast, the SGS and PC algorithms (Spirtes et al., 2000) given *no evidence* in favor of membership of X in $PC(T)$ and *no reliable proof* to the contrary, assumes that X has a common edge with T. IAMB (Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a) to the contrary, given *some* reliable evidence in favor of a variable X belonging to $MB(T)$ but *no reliable proof* to the contrary, outputs X as member of $MB(T)$ if X is in the tentative Markov blanket $TMB(T)$ and is agnostic with respect to membership in $MB(T)$ if X is outside $TMB(T)$. Bayesian scoring methods in small samples are dominated by their priors and typically they prefer sparse networks which lead to fewer false positives and more false negatives.

4.5. Reasons for Good Performance of Non-Symmetry Corrected Algorithms

The empirical evaluations in part I of this work (Aliferis et al., 2010) have shown that the addition of symmetry correction adds little to quality, while it detracts from computational efficiency. Evidently very often $EPC(T) \approx PC(T)$ in real-life distributions and targets of interest. In addition, due to imperfect power to detect and return strongly relevant features, applying symmetry correction leads to reduced power and increased false negatives.

4.6. Reasons for Good Performance of the $PC(T)$ Set Instead of the $MB(T)$ Set for Classification

According to the theoretical results summarized in Aliferis et al. (2010), under broad assumptions spouses are needed for optimal classification performance. Given that in the majority of data sets tested in Aliferis et al. (2010) as well as the experiments in Section 2 of the present paper, when the set of parents and children is used instead of $MB(T)$ it produces equal or almost equal performance, more compact feature sets and faster feature selection times than inducting the full $MB(T)$ (i.e., both $PC(T)$ and $MB(T)$ estimated under the same assumptions of the theory that predicts that $MB(T)$ is needed for optimal feature selection). In this sub-section we provide likely explanations for the empirically excellent performance of substituting the set $PC(T)$ in place of $MB(T)$ for classification (apart from the obvious possibility that spouses may be much fewer and with smaller predictive value than parents and children). Figure 3 describes visually five plausible scenarios explaining the phenomenon.

The first scenario corresponds to the situation whereby the target variable T does not have children (and thus no spouses) by virtue of domain constraints. Such situations happen when the target variable is a variable preceded in time by all other variables (e.g., patient outcome on the basis of earlier observations); or when naturally the target variable cannot have children (e.g., the target being meaning category of a text document as a function of patterns of presence/absence of words in the text). The second scenario describes the situation where a child is not observed (hidden) in the data set and thus the spouse B cannot be made informative for the target and thus it can neither be detected nor can it enhance a classifier built from the data. The third scenario describes the situation where a spouse has connecting paths to the target but these cannot be blocked simultaneously because of small sample size and/or choice of *max-k*. Hence GLL-PC could admit the spouse D as a member of $PC(T)$. The fourth scenario simply shows a case where a spouse is also a child (or parent) and thus will be a member of $PC(T)$ as well as $MB(T)$. Finally the fifth scenario shows that an unmeasured variable may make a spouse appear as having a direct edge to or from the target (and thus are detectable by GLL-PC).

We note that in practical data analysis and evaluations when both $PC(T)$ and $MB(T)$ are induced and are found to have similar classification performance, typically $MB(T)$ is much larger than $PC(T)$. However this may be a reflection of the inductive bias of GLL which prefers to admit potential false positives if they cannot be shown for sample size reasons to be independent of the target.

Finally note that explanations #1, 2, 3, and 4 are special cases of the assumptions of the Markov blanket induction theory and thus they do not refute these assumptions (whereas #5 violates causal sufficiency). In the discussion section we consider additional situations with violations of GLL assumptions.

4.7. Error Estimation Problems in Wrapping and Standard Filters Due to Small Sample Size. GLL Filtering is Less Sensitive to Error Estimation Difficulties and Robust to Small Samples

Wrapping has been praised as a feature selection methodology for its ability to tailor the feature selection to the inductive bias of the classifier(s) of choice as well as to the loss function of interest (Kohavi and John, 1997). Occasionally, this property will work against the analysis (see Section 7 for example for how it can jeopardize causal discovery). On the other hand, wrapping has been criticized for its very large computational

cost as well as on the grounds that it is subject to No Free Lunch Theorem limitations (i.e., a priori all wrappers are equally good, making it hard to find the right wrapper for the distribution, loss function and classifier(s) of interest) (Tsamardinos and Aliferis, 2003). In the present section we explain what we believe is perhaps the most serious practical shortcoming of wrapping feature selection methods, namely that *they rely on error estimation procedures that are often unreliable because of small sample sizes.* The difficulties that will be presented here help explain the sometimes poor performance of some of the feature selection algorithms in the evaluation part (Aliferis et al., 2010). In contrast, we will show that GLL filtering is resistant to these problems.

Recall that the critical point when applying error estimators is to have a sufficiently small variance and to be unbiased or to correct for any bias, as for example is the case of the (biased) Bootstrap estimator. Consider an idealized example where a greedy (steepest-descent) backward selection wrapper algorithm is applied on faithful data that contains 5 irrelevant features $I_1, \ldots, I_5$ and one strongly relevant feature S.

Assume that in reality the optimal feature set consisting of only the strongly relevant feature S gives a predictor model with true error measured by AUC is 0.75 in the large sample (i.e., in the distribution where the data is sampled from). For all practical unbiased error estimators, because of variability in the estimates of error due to small sample sizes, and because of potential sensitivity of the classifier employed to irrelevant features, some subsets that contain S will have error estimates in small sample situations that are larger and some smaller than the true AUC of 0.75. The backward wrapping starts by eliminating one variable at a time producing feature sets and corresponding predictor models and by eliminating the feature that decreases error the most relative to the starting model that contains all features. As a result, a feature set can be chosen, not because the error is truly decreased if we remove any more features, but because the error estimates vary and the backward wrapper (naively) does not take this into account. If the wrapper is configured to employ statistical significance tests each time it compares estimates of error between pairs of feature sets and corresponding classifiers, because statistical tests of error estimate differences are often underpowered (which is another manifestation of the large variance in error estimates) such tests will often fail to reveal true differences. Thus the wrapper can falsely conclude that two models have same error when in reality they do not. This will entail choosing wrongly the smallest of the two and eliminating valuable features. Also due to multiple comparisons, such an algorithm will falsely conclude for a proportion of feature sets that a difference in predictor model performance is statistically significant thus continuing removal of relevant features when they should not be removed.

We emphasize that this problem is not present in wrapper methods only. In traditional feature ranking methods, the above problem is also present but often ignored in the sense that many studies on feature ranking algorithms produce a performance-to-feature-number plot, with performance estimated on a single data set. However the practical data analysis problem of how to select a specific number of features that achieves at most some desired error is left unspecified and in fact subject to the same error estimation difficulty that applies to wrapping. Moreover, in recent algorithms such as RFE, the problem is acknowledged implicitly in the applied examples provided by the authors of the method, since feature sets are reduced by for example 50% in each iteration of the algorithm creating a new subset of features examined by cross-validation by the algorithm (Guyon et al., 2002). This is done to reduce overfitting of selected feature set to the data because of the large variability of error estimates. As evidenced by the evaluations presented in Aliferis et al. (2010), it is possible to improve on tradi-

Table 10: Trace of semi-interleaved HITON-PC without symmetry correction (i.e., GLL-PC-nonsym subroutine) showing insensitivity to error estimation difficulties that affect wrappers.

Action	Decision	Notes
Rank variables according to univariate association with target T	S (association = 0.8) I_1 (association = 0.3) I_2 (association = 0.1) I_3 (association = 0.1) I_4 (association = 0.05) I_5 (association = 0.0)	Some associations of irrelevant variables are non-zero due to sampling variation
Test S for inclusion: $\neg I(S, T)$	Admit S in $TPC(T)$	Assuming S is a strong predictor of the target, the power of the univariate test will be sufficient to reject independence
Test I_1 for inclusion: $I(I_1, T)$	Eliminate I_1	Test will be correct with probability 1-α (typically 0.95)
Test I_2 for inclusion: $I(I_2, T)$	Eliminate I_2	Test will be correct with probability 1-α (typically 0.95)
Test I_3 for inclusion: $\neg I(I_3, T)$	Consider I_3	Assume we were unlucky and had a false positive
Test I_3 for inclusion: $I(I_3, T \mid S)$	Eliminate I_3	Test will be correct with probability 1-α (typically 0.95). Very unlikely (probability = 0.0025) that I_3 will pass through second test
Test I_4 for inclusion: $I(I_4, T)$	Eliminate I_4	Test will be correct with probability 1-α (typically 0.95)
Test I_5 for inclusion: $I(I_5, T)$	Eliminate I_5	Test will be correct with probability 1-α (typically 0.95)
Test S for final elimination: no test to be made	Accept S	
Return $\{S\}$ as final output		

tional wrapping, ranking and RFE selection by applying statistical tests of difference of error estimates, or by increasing/decreasing the granularity of feature selection (i.e., proportion of features removed at each iteration). Still the produced feature sets are not optimal in parsimony. The numbers of strongly relevant, weakly relevant and irrelevant features is not critical to the existence of the problem, neither is the type of wrapper (forward, backward, forward-backward, GA, etc.) as long as some basic requirements are met: error estimation is not perfect but subject to sampling variability due to small sample, and enough features exist in data for enough error estimate comparisons to be spurious.

Contrary to the above, GLL filtering relies little on error estimation[3] and uses robust mechanisms to control false negatives and false positives separately for strongly relevant, weakly relevant and irrelevant features respectively. In Table 10 we give a concrete demonstration of how semi-interleaved HITON-PC (without symmetry correction for simplicity) is less prone to errors in the same example. The critical observation is for an irrelevant feature to enter $TPC(T)$ and stay in it, it has to survive multiple (i.e., $2^{|TPC(T)|}$)

3. Notice that some reliance on error estimation exists in domains where a suitable *max-k* and α are not known and need be optimized by cross-validation. The corresponding number of parameterizations is very small however (typically at the order of 10 combined parameter configurations) and thus error estimation is less likely to lead the algorithm astray. The same is true for the optional wrapping step in GLL-MB which selects features from a highly reduced set compared to the original feature set (notice that this wrapping step is seldom needed in practice and is reserved for higher sample settings).

Chunked Parallel GLL-PC Algorithm (not symmetry corrected)

Input: Dataset D, target variable T, desired number of data chunks ch.

1. Split the data D into ch arrays C_i of equal size, such that each array contains a non-overlapping subset of the variables plus T.
2. For all i, compute $ChunkPC_i(T) \leftarrow$ GLL-PC-nonsym(T, C_i)
3. $\boldsymbol{L} \leftarrow$ GLL-PC-nonsym(T, $\cup_i ChunkPC_i(T)$)
4. Return $\boldsymbol{L}$ and exit

Figure 4: Chunked Parallel GLL-PC algorithm (not symmetry corrected).

tests of conditional independence and each such test has probability $1 - \alpha$ to leave the irrelevant feature in $TPC(T)$. The total probability of failing to reject the irrelevant variable thus grows up to exponentially small to the number of tests performed and is independent of the sample size. In our simplified example with just one strongly irrelevant feature inside $TPC(T)$, each irrelevant feature has probability of entering and staying in $TPC(T)$ of at most $\alpha^2 = 0.0025$. This is true regardless of whether sample size is 10,000 samples or just 10 samples.

5. Algorithmic Extensions to GLL

In the present section we introduce algorithmic extensions to the Generalized Local Learning algorithms: parallel and distributed local learning and FDR pre-filtering.

5.1. Parallel and Distributed Local Learning

Following ideas for parallelizing the IAMB algorithm for $MB(T)$ estimation (Aliferis et al., 2002), we introduce a coarse-grain parallelization of GLL-PC that addresses two problems: (a) the data does not fit into fast memory (RAM), and (b) even if the data fits, we wish to speedup execution time by parallel processing. We allow for the possibility that the user may have access to just one node or, alternatively, may have access to several nodes arranged in a parallel cluster. The algorithm presented can return $PC(T)$ and can run with any instantiation of GLL-PC. The algorithm is designed to be correct provided that no symmetry correction is required (i.e., in distributions where $EPC(T) \equiv PC(T)$). Correct parallel/distributed versions in distributions where symmetry correction is needed can also be obtained as can algorithms that parallelize $MB(T)$ induction. In the present paper we only discuss parallel GLL-PC without symmetry correction because of its conceptual and implementation simplicity and speed, because it can be used for both causal discovery and prediction, and because as demonstrated empirically (Aliferis et al., 2010), many real distributions behave consistently with being "symmetrical" (i.e., $EPC(T) \equiv PC(T)$).

Chunked Parallel GLL-PC algorithm (not symmetry corrected): This algorithm assumes that one has access to several nodes and that the data can fit to the available memory once distributed, while it may or may not fit to a single node. Initially the algorithm divides the input data D into ch chunks C_i such that every C_i includes all cases, but only a subset $\boldsymbol{V}_i$ of the variable set $\boldsymbol{V}$ plus T. For simplicity we assume that each chunk has an equal number of features (that can be determined, for example, by the maximum size that can be processed in fast memory or the number of available

computer nodes in a parallel implementation). Variations where unequal variable allocations are employed can be easily obtained in similar fashion. Then GLL-PC-nonsym is run on each chunk (as indicated by the extra input argument C_i) returning $ChunkPC_i(T)$ (i.e., parents and children of T in chunk C_i). Next, GLL-PC-nonsym is run on one node with the union $\cup_i ChunkPC_i(T)$, it obtains a local neighborhood $\boldsymbol{L}$, and terminates by outputting $\boldsymbol{L}$. Figure 4 gives the parallel GLL-PC high-level pseudo-code. Step #2 is the parallel step.

We note that a potential problem with chunked GLL-PC is that the tentative neighborhood in some chunk(s) may grow very large (up to the size of the chunk in the worst case) while the true neighborhood across all variables may be very small. This creates the possibility of overflow both in the sense of data not fitting in a single node and in the sense of not having enough sample size to perform reliable statistical inferences.

Theorem 1 *Chunked parallel GLL-PC without symmetry correction is sound given the sufficient conditions for soundness of GLL-PC and the requirement that in the generating distribution P, $PC(T)$ is the same as the Extended $PC(T)$ (see definition of $EPC(T)$ in Aliferis et al. 2010).*

Proof In each chunk, GLL-PC-nonsym will identify all true members of $PC(T)$ that are in the chunk (because these can never be rendered independent of T, according to Theorem 1 in Aliferis et al. 2010) and some false positives which cannot be eliminated without conditioning on $PC(T)$ members that belong to another chunk. Thus in step #3, GLL-PC-nonsym is executed on a superset of $PC(T)$. By definition, all non-members of $PC(T)$ can be rendered independent of T conditioned on some subset of $PC(T)$ as long as $PC(T) \equiv EPC(T)$. Since $PC(T) \equiv EPC(T)$, the identified $PC(T)$ will be correct. ■

The complexity of Chunked Parallel GLL-PC without symmetry correction is in the worst case exponentially slower than running GLL-PC on all data. This is because the complexity of GLL-PC is worst-case exponential to the size of $TPC(T)$ and while $TPC(T)$ in all data can be very small, in some chunks $TPC(T)$ can be as large as the chunk itself. When however local neighborhoods in each chunk are smaller than the global $TPC(T)$ and since GLL-PC is worst-case exponential, the algorithm can also be exponentially faster than running GLL-PC on all data. This is in sharp contrast with parallel IAMB where both the speedup is linear to the number of chunks in the best case (upper bound on the speed-up factor is *ch*) and worst-case running time is a small constant multiple of running the algorithm on all data (Aliferis et al., 2002).

Chunked Distributed GLL: When we run the algorithm with data already distributed, the data splitting and transfer step #1 (as well as associated transfer cost) is omitted. Typically we will need to link the distributed data using a suitable common key. For example consider a large organization wishing to analyze data in order to find determinants of production costs overall many and geographically dispersed branches, each with its own local data set and different recorded features. An appropriate key might be time label of observations. Another example is hospital patient data distributed among numerous local databases in different units and labs of the hospital, where patient id is a suitable key.

Chunked GLL with single CPU: This variant assumes access to one CPU only and addresses the problem of data not fitting in the fast memory. By processing parts of the data sequentially and obtaining a small superset of $PC(T)$ each time, a much larger data set than what fits in fast memory can be analyzed.

We now apply a parallel version of semi-interleaved HITON-PC on the four largest real data sets (*Ohsumed*, *ACPJ_Etiology*, *Thrombin*, and *Nova*) of the empirical evaluation

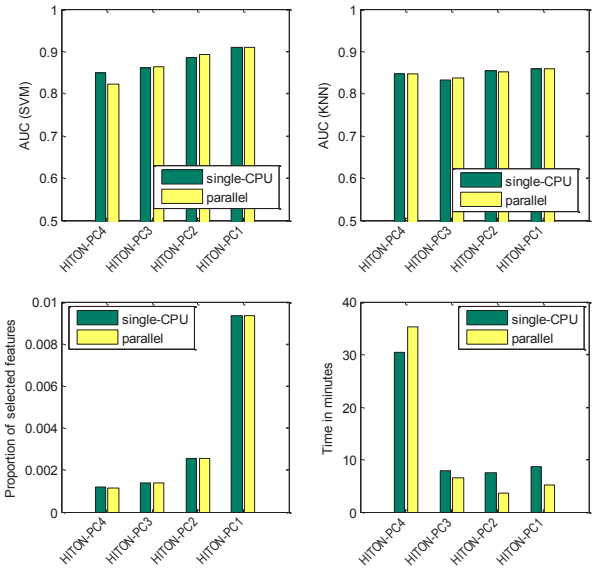


Figure 5: Results of application of single-CPU and parallel versions of semi-interleaved HITON-PC on the four largest real data sets (*Ohsumed*, *ACPJ_Etiology*, *Thrombin*, and *Nova*). Average results over 4 data sets are shown. The following versions of HITON-PC are used: HITON-PC4 (max-k=4, α=0.05), HITON-PC3 (max-k=3, α=0.05), HITON-PC2 (max-k=2, α=0.05), HITON-PC1 (max-k=1, α=0.05).

in Aliferis et al. (2010). We use 10 CPU's on the ACCRE cluster described in Aliferis et al. (2010). As can be seen in Figure 5 the parallel version achieves the same parsimony and classification performance as the single-CPU application with speedup for three out of four versions of HITON-PC (see Figure 5). P-values from the permutation test of the null hypothesis that single-CPU and parallel GLL-PC algorithms achieve the same performance are 0.7468 (for SVM classification), 0.4950 (for KNN classification), 0.2408 (for proportion of selected features), and 0.6374 (for running time in minutes). We note that running times for HITON-PC algorithm in this subsection are less than in the remainder of the paper because these experiments were executed on the most recent version of the ACCRE cluster.

5.2. FDR pre-Filtering

As explained in Section 3, in simulated and resimulated data sets with weak-signal/small sample and in all-irrelevant features situations, removing features using false discovery rate control can improve the number of false positives in HITON-PC and MMPC. We applied HITON-PC with FDR pre-filtering in all real data sets of Aliferis et al. (2010). As can be seen in Figure 6, this enhancement does not entail improvements in parsimony, classification performance or running time in the data sets tested. P-values from the permutation test of the null hypothesis that GLL-PC algorithms with and without FDR correction achieve the same performance are 0.5254 (for SVM classification),

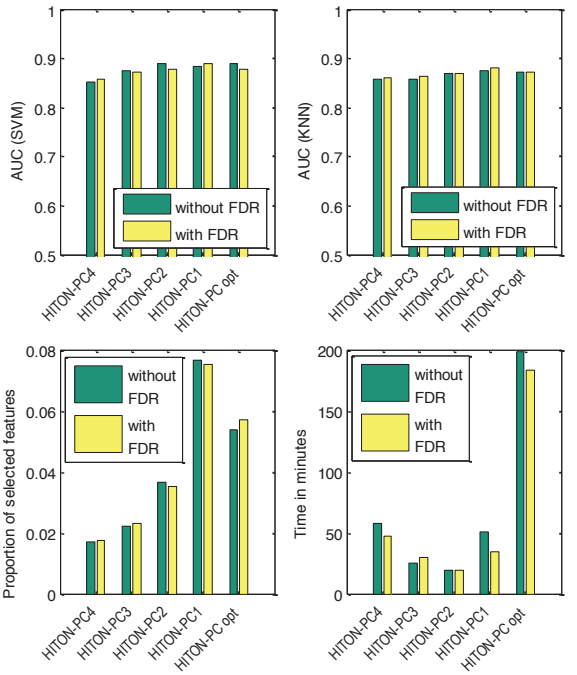


Figure 6: Results of application of semi-interleaved HITON-PC with and without FDR correction on 13 real data sets. Average results over the data sets are shown. The following versions of HITON-PC are used: HITON-PC4 (*max-k*=4, α=0.05), HITON-PC3 (*max-k*=3, α=0.05), HITON-PC2 (*max-k*=2, α=0.05), HITON-PC1 (*max-k*=1, α=0.05), HITON-PC opt (*max-k* and α are optimized over values $\{1, 2, 3, 4\}$ and $\{0.05, 0.01\}$, respectively, by cross-validation to maximize SVM classification performance).

0.3698 (for KNN classification), 0.9426 (for proportion of selected features), and 0.3776 (for running time in minutes). Since however the algorithm exhibits small sensitivity to false positives due to multiple comparisons when many irrelevant features are expected and few relevant features are present, we recommend pre-filtering with FDR. Alternatively, if one gets a few variables combined with error estimates consistent with uninformative classifier, then re-running standard GLL with FDR pre-processing can be tried.

When evaluating local causal discovery performance in the simulated data of Aliferis et al. (2010), semi-interleaved HITON-PC with FDR pre-processing achieves dramatically better performance than other algorithms including other HITON and MMPC variants with respect to graph distance score, which indicates average causal proximity to the target of the returned variables. Specifically, in large sample (N=5,000) HITON-PC with FDR correction achieves up to 5-fold reduction in the graph distance score relative to the best non-FDR filtered causal algorithm and up to 9-fold reduction compared to the best non-causal algorithm. In small sample (N=200) the reduction in both cases is 2-fold. P-values from the permutation test of the null hypothesis that

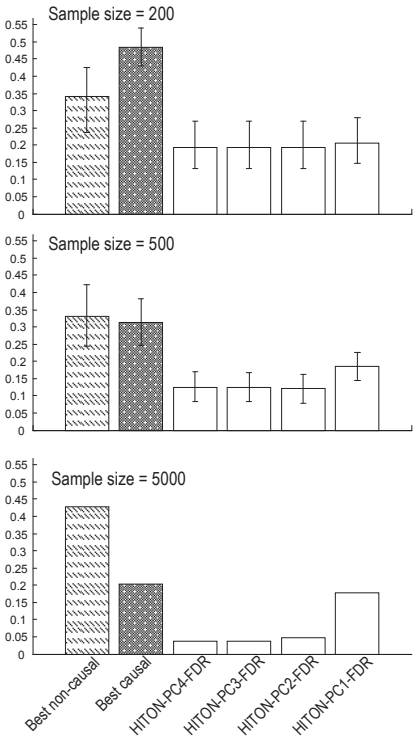


Figure 7: Graph distances averaged over all 9 simulated and resimulated data sets, all selected targets in each data set, and multiple samples of a given size. The following versions of semi-interleaved HITON-PC with FDR correction are used: HITON-PC4-FDR (*max-k*=4, α=0.05), HITON-PC3-FDR (*max-k*=3, α=0.05), HITON-PC2-FDR (*max-k*=2, α=0.05), and HITON-PC1-FDR (*max-k*=1, α=0.05). "Best causal" is the best causal feature selection algorithm among techniques that do not incorporate FDR. "Best non-causal" is the best non-causal feature selection algorithm. See Aliferis et al. (2010) for a detailed list of algorithms.

the best non-causal algorithm performs the same as the average HITON-PC with FDR correction are $<$0.0001 for sample sizes 200, 500, and 5,000. P-values for comparison with the best causal algorithm are $<$0.0001, 0.0030, and $<$0.0001 for sample sizes 200, 500, and 5000, respectively. See Figure 7. This improvement incurs only a very small decrease in sensitivity as evidenced by small concurrent increases in false negatives.

6. Spanning Local to Global Learning

In the present section we investigate the use of local learning methods (such as GLL) for global learning in a divide-and-conquer fashion. We remind that a major motivation for pursuing local causal learning methods is scaling up causal discovery and causal feature selection as explained in Aliferis et al. (2010). Although similar concepts can

be used for region learning, we will not address this type of discovery problem here. The main points of the present section are that (a) the local-to-global framework can be instantiated in several ways with excellent empirical results; (b) an important previously unnoticed factor is the variable order in which to execute local learning, and (c) trying to use non-causal feature selection in order to facilitate global learning (instead of causal local learning) is not as a promising strategy as previously thought.

LGL: Local-to-Global Learning

1. Find *PC(X)* for every variable *X* in the data using an admissible instantiation of GLL-PC and prioritizing which variables to induce *PC(X)* for, according to a prioritization strategy.
2. Piece together the undirected skeleton from the local GLL-PC results.
3. Use any desired arc orientation scheme to orient edges.

Figure 8: Local-to-Global Learning (LGL) algorithmic schema.

MMHC Global Learning Algorithm

1. Find *PC(X)* for every variable *X* in data using MMPC (without symmetry correction) and lexicographic prioritization.
2. Piece together the undirected skeleton using an "OR rule" (an edge exists between *A* and *B* iff *A* is in *PC*(*B*) or *B* is in *PC*(*A*)).
3. Use greedy steepest-ascent TABU search and BDeu score to orient edges.

Figure 9: MMHC global learning algorithm as an instance of LGL.

HHC Global Learning Algorithm

1. Find *PC*(*X*) for every variable *X* in data using semi-interleaved HITON-PC (without symmetry correction) and lexicographic prioritization.
2. Piece together the undirected skeleton using an "OR rule" (an edge exists between *A* and *B* iff *A* is in *PC*(*B*) or *B* is in *PC*(*A*)).
3. Use greedy steepest-ascent TABU search and BDeu score to orient edges.

Figure 10: HHC global learning algorithm as an instance of LGL.

6.1. General Concepts

A precursor to the main idea behind the local-to-global learning approach can be found in SCA (Friedman et al., 1999), where a heuristic approximation of the local causes of every variable constraints the space of search of the standard greedy search-and-score Bayesian algorithm for global learning increasing thus computational efficiency. Given powerful methods for finding local neighborhoods, provided by the GLL framework, one can circumvent the need for uniform connectivity (as well as user knowledge of that connectivity) and avoid the application of inefficient heuristics employed in SCA thus improving on quality and speed of execution. Figure 8 provides the general algorithmic schema term LGL (for local-to-global learning). Steps #1–3 can be instantiated in

numerous ways. If an admissible GLL-PC (as defined in Section 4 of Aliferis et al. 2010) is used in step #1, and step #2 is consistent with the results of GLL-PC for all variables, and a sound orientation scheme in step #3, then the total algorithm is trivially sound under the assumptions of correctness of GLL-PC. These are the admissibility requirements for the LGL template. It follows that:

Proposition 1 *Under the following sufficient conditions we obtain correctly oriented causal graph with any* admissible *instantiation of LGL:*

a. There is a causal Bayesian network faithful to the data distribution P;

b. The determination of variable independence from the sample data D is correct;

c. Causal sufficiency in $\boldsymbol{V}$.

The recently-introduced algorithm MMHC is an instance of the LGL framework (Tsamardinos et al., 2006). Figure 9 shows how MMHC instantiates LGL. MMHC is not sound with respect to orientation because greedy steepest-ascent search is not a sound search strategy for search-and-score global learning. Despite being theoretically not sound the algorithm works very well in practice and in an extensive empirical evaluation it was shown to outperform in speed and quality several state-of-the-art algorithms (Greedy Search, GES, OR, PC, TPDA, and SCA) (Tsamardinos et al., 2006).

6.2. A New Instantiation of LGL: HHC

To demonstrate the generality and robustness of the LGL framework we provide here as an instantiation of LGL, a new global learning algorithm termed HHC (see Figure 10), and compare it empirically to the state-of-the-art MMHC algorithm. We also show that the two algorithms are not identical in edge quality or computational efficiency, with the new algorithm being at least as good on average as MMHC.

Table 11 presents results for missing/extra edges in undirected skeleton, number of statistical tests for construction of skeleton, structural Hamming distance (SHD), Bayesian score, and execution time on 9 of the largest data sets used for the evaluation of MMHC. Since the data sets were simulated from known networks, the algorithm output can be compared to the true structure. As can be seen, in all 9 data sets, HHC performs equally well with MMHC in terms of SHD and Bayesian score. In 8 out of 9 data sets it performs from 10% to 50% fewer tests, and in one data set (*Link*) it performs $>$10 times the tests performed by MMHC resulting in running 35% slower in terms of execution time. Because MMHC was found to be superior to a number of other algorithms for the data sets tested, HHC's better performance over MMHC in 8 out of 9 data sets (in terms of number of statistical tests for skeleton construction) and similar performance in 9 out of 9 data sets (in terms of quality metrics) translates also to excellent performance of HHC relative to Greedy Search, GES, OR, PC, TPDA, and SCA (Tsamardinos et al., 2006).

6.3. Importance of Variable Prioritization for Quality and Efficiency

An important parameter of local-to-global learning previously unnoticed in algorithms such as SCA and MMHC is the ordering of variables when executing the local causal discovery variable-by-variable (i.e., not in parallel). We will assume that results are shared among local learning runs of GLL-PC, that is when we start learning $PC(X)$ by

Table 11: Comparison of HHC and MMHC global learning algorithms. Both algorithms were executed on a random sample of size 1000, using default parameters of MMHC as implemented in *Causal Explorer* (i.e., G^2 test for conditional independence, $\alpha = 0.05$, *max-k* = 10, Dirichlet weight = 10, BDeu priors).

HHC

	Dataset								
	Child10	*Insurance10*	*Alarm10*	*Hailfinder10*	*Pigs*	*Munin*	*Lung_Cancer*	*Gene*	*Link*
Extra edges in learned skeleton	95	143	176	1265	276	36	621	601	1456
Missing edges in learned skeleton	25	149	165	359	0	257	91	6	439
Structural Hamming distance for DAG	101	297	344	728	4	273	187	72	1150
Bayesian score for DAG	-188.61	-229.02	-178.56	-738.77	-496.11	-33.14	-559.43	-651.36	-337.74
Number of statistical tests for skeleton construction	28,879	52,757	82,543	217,490	134,244	733	859,348	401,779	7,931,044
Time for building skeleton (in minutes)	0.74	1.59	2.47	8.05	3.98	0.23	24.40	12.32	537.72
Total time for running algorithm (in minutes)	1.21	3.32	6.80	24.84	14.33	0.47	181.97	60.14	563.46

MMHC

	Dataset								
	Child10	*Insurance10*	*Alarm10*	*Hailfinder10*	*Pigs*	*Munin*	*Lung_Cancer*	*Gene*	*Link*
Extra edges in learned skeleton	71	128	184	1220	281	38	567	557	1541
Missing edges in learned skeleton	25	148	164	352	0	258	88	4	396
Structural Hamming distance for DAG	100	296	346	725	4	275	191	69	1145
Bayesian score for DAG	-188.95	-229.03	-179.09	-738.80	-496.11	-33.12	-559.01	-651.12	-337.62
Number of statistical tests for skeleton construction	32,980	67,943	90,117	243,571	177,278	1,023	1,360,493	451,364	644,055
Time for building skeleton (in minutes)	0.81	1.99	2.49	12.81	5.45	0.38	55.16	12.23	382.93
Total time for running algorithm (in minutes)	1.42	3.79	5.21	29.54	13.11	0.46	451.70	51.84	415.69

GLL-PC rather than starting with an empty $TPC(X)$ set, we start with all variables Y: $X \in PC(Y)$. This constitutes a sound instantiation of the GLL-PC algorithm template as explained in Aliferis et al. (2010). Figure 11 gives two extreme examples where the right order can "make-or-break" an LGL algorithm.

In Figure 11(a) it is straightforward (and left to the reader to verify) that an order of local learning $< X_1, X_2, \ldots, X_{100}, Y >$ without symmetry correction (the latter being a

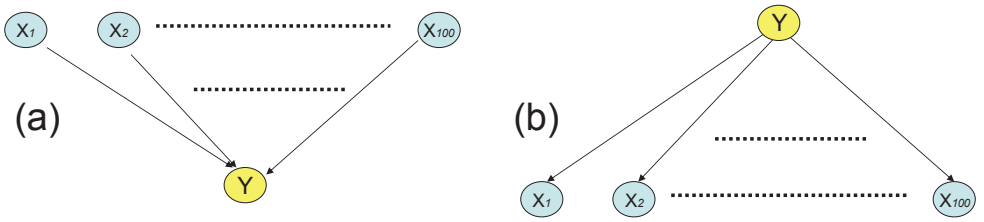


Figure 11: Two examples where the variable ordering for local learning can make execution of the LGL algorithm from quadratic to exponential-time.

reasonable choice as we have seen) requires a quadratic number of conditional independence tests (CITs) for the unoriented graph to be correctly learned. However, the order of local learning $< Y, X_1, X_2, \ldots, X_{100} >$ requires up to an exponential number of CITs as *max-k* and sample are allowed to grow without bounds. Even with modest *max-k* values, the number of CITs is higher-order polynomial and thus intractable. Even when Y is not in the beginning but as long as a non-trivial number of X's are after it in the ordering, the algorithm will be intractable or at least very slow. The latter setting occurs in the majority of runs of the algorithm with random orderings.

In Table 12 we provide data from a simulation experiment showing the above in concrete terms and exploring the effects of limited sample and connectivity at the same time. As can be seen, under fixed sample, running HHC with order from larger to smaller connectivity, as long as the sample is enough for the number of parents to be learned (i.e., number of parents is ≤ 20), increases run time by more than 100-fold. However because sample is fixed, as the number of parents grows the number of conditional independence tests equalizes between the two strategies because CITs that have too large conditioning sets for the fixed sample size are not executed. Although the number of CITs is self-limiting under these conditions, quality (in terms of number of missing edges, that is, number of undiscovered parents of T) drops very fast as the number of parents increases. The random ordering strategy trades off quality for execution time with the wrong (larger-to-smaller connectivity) ordering, however in all instances the right ordering offers better quality and 2 to 100-fold faster execution that random ordering.

A more dramatic difference exists for the structure in Figure 11(b) where Y is a parent of all X's. Here the number of tests required to find the parent (Y) of each X_i is quadratic to the number of variables with the right ordering (low-to-high connectivity) whereas an exponential number is needed with the wrong ordering (large-to-small connectivity). Because the sample requirements are constant to the number of children of Y, quality is affected very little and there is no self-restricting effect of the number of CITs, opposite to what holds for causal structure in Figure 11(a). Hence the number of CITs grows exponentially larger for the large-to-small connectivity ordering versus the opposite ordering and a similar trend is also present for the average random ordering in full concordance with our theoretical expectations. See Table 13 for results of related simulation experiments.

These results show that in some cases, *it is possible to transform an intractable local learning problem into a tractable one by employing a global learning strategy (i.e., by exploiting*

Table 12: Results of simulation experiment with HHC algorithm. The graphical structure is depicted on Figure 11(a). HHC was run on a random sample of size 1,000 with G^2 test for conditional independence, α=0.05, *max-k* = 5, Dirichlet weight = 10, BDeu priors.

	order from low-to-high connectivity			random order (average results over 10 orders)			order from high-to-low connectivity		
Number of parents of Y	extra edges	missing edges	CITs	extra edges	missing edges	CITs	extra edges	missing edges	CITs
10	2	0	63	2	0	2,461	2	0	4,325
20	4	0	233	4.7	5.2	26,203	5	7	29,774
30	12	0	526	12	12.4	41,499	11	21	9,020
40	13	0	904	16.4	20.1	51,269	19	33	5,626
50	22	7	1,428	28.8	30	16,828	34	43	4,149
60	29	7	2,001	32.9	35.7	36,950	38	54	3,862
70	41	19	2,773	45.7	37.9	24,456	55	63	4,464
80	58	28	3,652	65.4	55.1	12,630	70	74	5,023
90	66	35	4,634	72.3	57.6	16,718	87	85	5,592
100	77	44	5,594	88.7	80	16,266	96	94	7,229

Table 13: Results of simulation experiment with HHC algorithm. The graphical structure is depicted on Figure 11(b). HHC was run on a random sample of size 1,000 with G^2 test for conditional independence, α=0.05, *max-k*=5, Dirichlet weight = 10, BDeu priors. Empty cells correspond to experiments when the algorithm did not terminate within 10,000,000 CITs.

	order from low-to-high connectivity			random order (average results over 10 orders)			order from high-to-low connectivity		
Number of children of Y	extra edges	missing edges	CITs	extra edges	missing edges	CITs	extra edges	missing edges	CITs
10	1	0	106	1	0	2,342	1	0	4,366
20	11	0	489	9.7	0	141,148	9	0	377,448
30	18	0	1,173	16.8	0	2,321,030	17	0	5,020,400
40	24	0	1,968	-	-	-	-	-	-
50	33	0	3,190	-	-	-	-	-	-
60	48	0	5,031	-	-	-	-	-	-
70	53	0	6,899	-	-	-	-	-	-
80	71	0	8,939	-	-	-	-	-	-
90	76	0	11,448	-	-	-	-	-	-
100	95	0	14,677	-	-	-	-	-	-

asymmetries in connectivity). Thus the variable order in local-to-global learning may have promise for substantial speedup and improved quality in real-life data sets (assuming the order of connectivity is known or can be estimated). However the optimal order is a priori unknown for some domain. Can we use local variable connectivity as a proxy to optimal order in real data? The next experiment assumes the existence of an oracle that gives the true local connectivity for each variable. The experiment examines empirically the effect of three orders (low-to-high connectivity, lexicographical (random) order, and

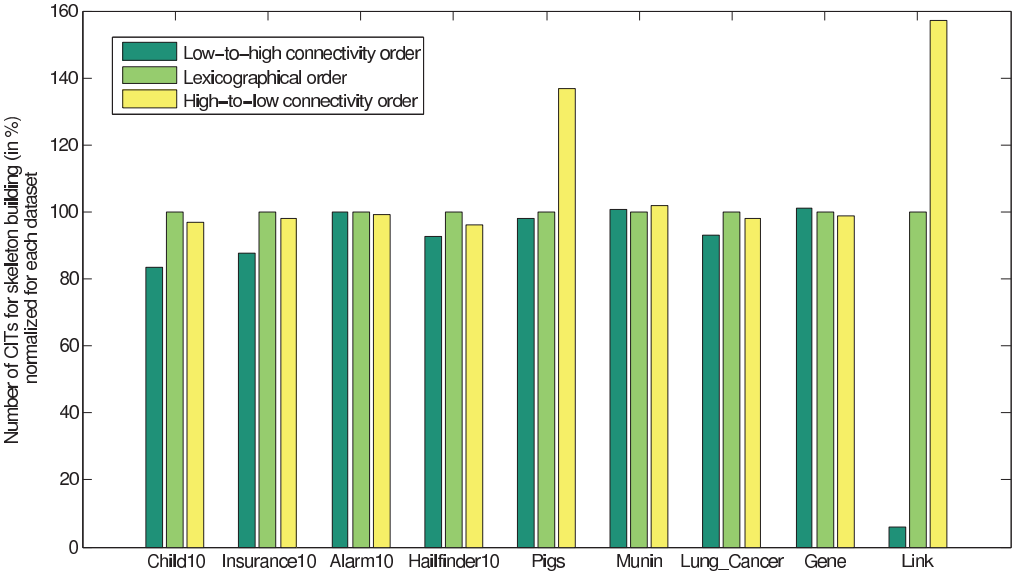


Figure 12: Number of CITs required for skeleton construction during execution of HHC expressed as % points and normalized within each data set to lexicographical order. Data for three orderings of variables is shown on the figure: low-to-high connectivity, lexicographical, and high-to-low connectivity orders. HHC was executed with same parameters as in Table 11. More detailed results are provided in Table 11 and Table S21 in the online supplement.

high-to-low connectivity order) on the quality of learning and number of CITs in the MMHC evaluation data sets. It also compares the sensitivity of HHC to order.

As can be seen in Figure 12, the order does have an effect on computational efficiency however not nearly as dramatic in the majority of these more realistic data sets compared to the simpler structures of Figure 11. An exception is the *Link* data set in which low-to-high connectivity allows HHC to run 17 times faster than lexicographical (random) order and 27 times faster than high-to-low connectivity order. For the majority of cases, running these algorithms with lexicographical (i.e., random) order is very robust and does not affect quality adversely but affects run time and number of CITs to a small degree (details in Table S21 in the online supplement).

Thus, while connectivity affects which variable order is optimal in LGL algorithms, ranking by local connectivity does not exactly correspond to the optimal order. Figure S3 in the online supplement shows the number of CITs plotted against true local connectivity in each one of the 9 data sets used in this section. Related to the above, Figure S4 in the supplement also shows the distribution of true local connectivity in each data set. Consistent trends indicating the shape of the distributions by which the degree of local connectivity may determine an advantage of orderings low-to-high to high-to-low connectivity are not apparent in these data sets.

We hypothesize that more robust criteria for the effect of variable ordering in LGL algorithms can be devised. For example, the number or total cost of CITs required to locally learn the neighborhood of each variable. Such criteria are also more likely to

Table 14: Results for hybrid methods using RFE, LARS-EN and UAF.

	Child10				*Pigs*				*Hailfinder10*			
	RFE	***LARS***	***UAF***	***HHC***	***RFE***	***LARS***	***UAF***	***HHC***	***RFE***	***LARS***	***UAF***	***HHC***
Extra edges in learned skeleton	2078	7558	3014	95	2262	29570	5593	276	6424	40948	7904	1265
Missing edges in learned skeleton	26	8	20	25	2	0	0	0	461	211	325	359
Structural Hamming distance for DAG	121	117	135	101	76	102	7	4	796	756	733	728
Bayesian score for DAG	-190.0	-189.1	-189.8	-188.61	-497.2	-496.8	-496.4	-496.11	-740.5	-736.4	-737.4	-738.77
Time for building skeleton (in minutes)	41.63	43.57	44.97	0.74	348.44	184.47	355.59	3.98	572.13	365.45	581.34	8.05
Total time for running algorithm (in minutes)	43.23	48.52	47.05	1.21	361.15	265.07	373.54	14.33	603.62	503.63	612.63	24.84

	Gene				*Lung Cancer*			
	RFE	***LARS***	***UAF***	***HHC***	***RFE***	***LARS***	***UAF***	***HHC***
Extra edges in learned skeleton	4039	55384	9834	621	7469	38753	12486	601
Missing edges in learned skeleton	47	8	28	91	120	24	78	6
Structural Hamming distance for DAG	125	156	115	187	220	139	175	72
Bayesian score for DAG	-658.3	-653.1	-655.1	-559.43	-562.4	-555.6	-560.1	-651.36
Time for building skeleton (in minutes)	737.99	513.12	783.97	24.40	493.84	377.85	563.46	12.32
Total time for running algorithm (in minutes)	784.54	912.33	890.63	181.97	708.77	1096.19	855.18	60.14

be available or to be approximated well during practical execution of an algorithm than true connectivity. A variant of HHC, algorithm HHC-OO (standing for HHC with optimal order) (Aliferis and Statnikov, 2008) orders variables dynamically according to heuristic approximations to the total number of CITs for each variable. We also conjecture that the strategy for piecing together the local learning results strongly interacts with the local variable ordering to determine the tradeoff between the quality and efficiency of LGL algorithms. Evaluation of these hypotheses is outside the scope of the present paper.

6.4. Using non-Causal Feature Selection for Global Learning

In recent years several researchers have proposed that because modern feature selection methods can deal with large dimensionality/small sample data sets, they could also be used to speed up or approximate large scale causal discovery (e.g., Kohane et al. 2003 use univariate feature selection to build so-called "relevance networks"), or hybrid methods can be employed that use feature selection as a pre-processing to build a skeleton and then an orientation algorithm like Greedy Search in the spirit of MMHC and LGL (Schmidt et al., 2007). The results of Aliferis et al. (2010) contradict this postulate because they show that non-causal feature selection does not give locally correct results.

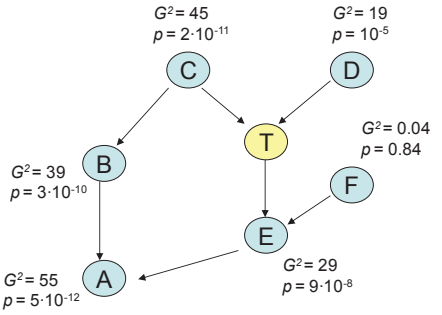
However it is still conceivable that orientation-and-repair post-processing algorithms (e.g., with Bayesian search-and-score) can still provide a high quality final causal graph. We test this hypothesis by examining several such hybrid methods using respectively RFE, LARS-EN and UAF post-processed by Greedy TABU Bayesian search-and-score. We use simulated data sets from 5 out of 9 Bayesian networks employed earlier in the present section. This is because the other 4 networks cannot be used for reliable training and testing of the underlying classifier since they have several variables with very unbalanced distributions. As shown in Table 14, the hypothesis is not corroborated by the experimental results. In particular, Greedy Search with feature selection-based skeleton, exhibits substantial drops in quality of the returned networks (measured by structural hamming distance Tsamardinos et al., 2006) and typically more than one order of magnitude longer running times compared to HHC with lexicographical (random) variable ordering. On the basis of these findings, which are consistent with the results in Aliferis et al. (2010), we do not find encouraging evidence that non-causal feature selection can be used as an adjunct to global causal discovery. Strong evidence exists however in favor of using principled local causal methods instead, within the frameworks of LGL.

7. Using Causal Graphs and Markov Blanket Theory as a Conceptual Analysis Framework for Feature Selection Methods

In the present section we show that by adopting a causal structural perspective founded on the theoretical results outlined in Aliferis et al. (2010), several strengths and weaknesses and general performance characteristics of non-causal feature selection algorithms become apparent and our empirical findings in Aliferis et al. (2010) can be better understood. We review several established and state-of-the-art methods both from a feature selection perspective (e.g., does the algorithm exhibit false positives and false negatives relative to minimal feature set that yields optimal predictivity?) and from a causal discovery perspective (is the output of the algorithm causally sound?). With respect to the latter for reasons elucidated in Aliferis et al. (2010), we focus on localization of causal inferences (i.e., whether the feature selection output is locally causally correct), and when this is not obtained, we examine whether some other useful causal inference can be made.

7.1. Univariate Association Filtering

Figure 13 shows the causal structure of a data-generating process. The causal structure is parameterized as shown in Appendix Figure 19. This structure and parameterization entails that $association(B,T) < association(C,T)$. Because of *synthesis of information along two paths* however, $association(A,T) > association(C,T)$ and $association(A,T) > association(E,T)$. The example illustrates that from the feature selection perspective the optimal predictor set (i.e., the Markov blanket) for predicting or classifying the target T is $\{C,D,E,F\}$. However, because univariate associations of non-$MB(T)$ members can be higher than those of members, false positives are incurred when selecting features using univariate association-based filters. Furthermore, spouses without connecting path to the target will have zero univariate association and thus will not be selected at all by univariate filtering. The embedded table shows the false positives and false negatives (relative to the gold standard set $MB(T)$) at each possible threshold for variable inclusion. In all cases predictivity is suboptimal.



Select first *k* features:

Selected		FN	FP	Predictivity
A	(k=1)	C,D,E,F	A	Suboptimal
A,C	(k=2)	D,E,F	A	Suboptimal
A,B,C	(k=3)	D,E,F	A,B	Suboptimal
A,B,C,E	(k=4)	D,F	A,B	Suboptimal
A,B,C,D,E	(k=5)	F	A,B	Suboptimal

Figure 13: Limitations of univariate feature selection explained using a causal graph perspective. Strength of univariate association with the target variable T is measured in a fixed sample of size 10,000 by the negative p-value of a G^2-test and depicted next to each variable.

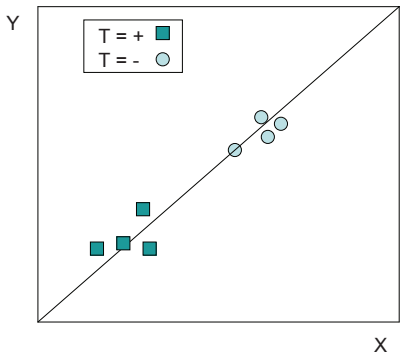


Figure 14: Example showing that Principal Component Analysis yields redundant features.

From the causal discovery perspective, the example makes evident that non-causally relevant features such as A and B can be selected with higher ranking than causally relevant ones such as D and E. Association synthesis thus forbids an interpretation of the higher-ranked causal variables as more direct causes (or effects) than lower-ranked features even when all of them are causal. Worse yet, even without synthesis, an arbitrarily large number of non-causal features can be selected before truly causal ones are selected. To see why this is the case consider that between C and B there may be arbitrarily many variables arranged in a chain so that their association with T is larger than that of both true cause D and true effect E.

7.2. Principal Component Analysis

As can be seen in Figure 14, the principal component defined by the diagonal ($Y - X = 0$) perfectly separates the two target classes and will be chosen by a PCA procedure since

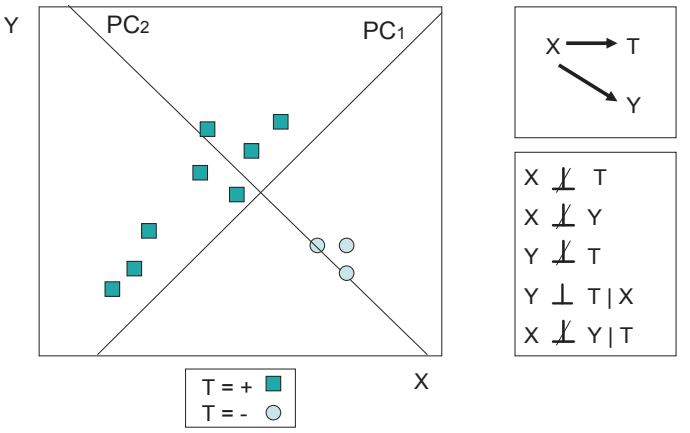


Figure 15: Example showing that Principal Component Analysis yields locally causally inconsistent results.

it explains maximum proportion of variance in the data. While projecting the original data on this single dimension reduces dimensionality of the classification problem, from the perspective of finding the original features that are important and non-redundant the method leads to false positives (since the coefficients of both Y and X are equal in the depicted Principal Component, indicating that both features are deemed equally necessary).

The example in Figure 15 shows that PCA is not sound for causal discovery. As shown in the figure, X is a direct cause of T and Y is not causal for T but confounded by X. Application of causal learning via the usual assumptions and procedures reveals that X is a direct cause or effect of T and that Y is not directly causally linked with T (the requisite conditional independence tests are depicted). However, an optimal procedure for Principal Component classification will select the second principal component PC_2 which achieves perfect classification. However both X and Y have equal coefficients in each principal component. Hence PCA may select both redundant features and non-causal features.

7.3. Feature Selection Using SVM Weights

A fundamental weakness of the maximum-gap inductive bias, as employed in SVMs, is its local causal inconsistency. Consider a scenario (Figure 16) similar to the previous sub-section where we wish to discover the direct causes of a response variable T, from observations about variables X, Y, T. Assume for simplicity that T is a terminal variable and thus X and Y precede it in time. For example, T can be a clinical phenotype and X, Y can be gene expression values. The causal process that generates the data is seen in the upper right corner of Figure 16. As can be seen in the left part of the figure, the SVM classifier can perfectly predict T using X and Y as predictors. In doing so it prefers the classifier with gap G1 to the classifier with smaller gap G2. The preferred classifier assigns non-zero (and in fact equal) weights to both X, Y thereby admitting Y in the local causal neighborhood if selected variables are interpreted causally. However, X *renders Y independent from T and not vice versa.* More generally, in distributions where

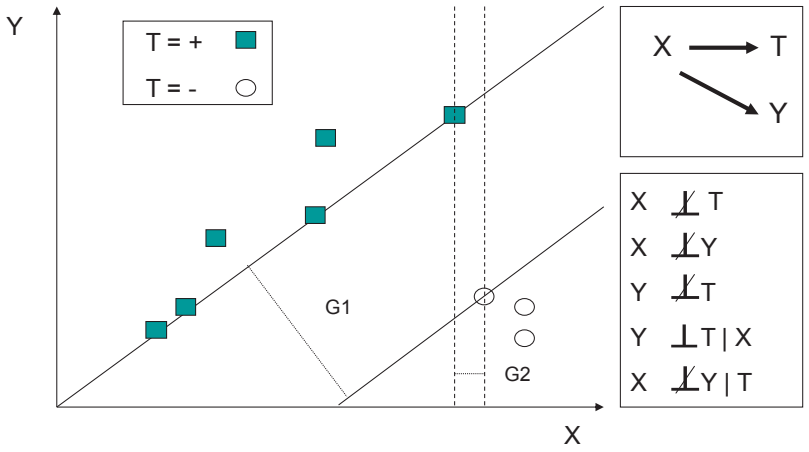


Figure 16: Example showing that SVM weight-based feature selection yields locally causally inconsistent results and redundant features.

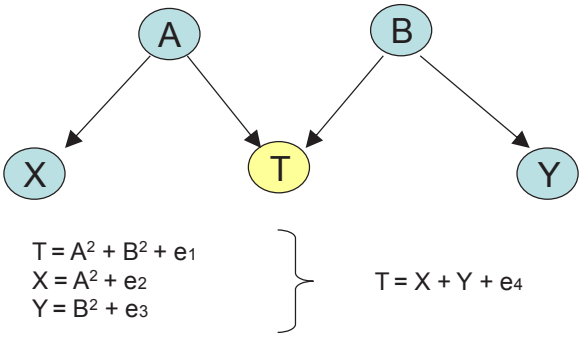
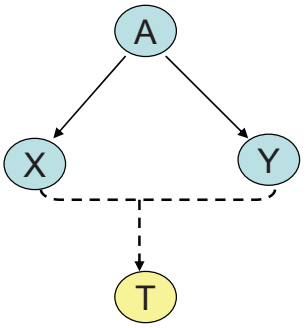


Figure 17: Example showing that wrapping, by tailoring feature selection to the classifier inductive bias may produce causally misleading results.

the Causal Markov Condition holds, SVMs will occasionally fail to detect that Y is not a local cause of T. Sound causal discovery algorithms do not face this problem, however. In addition, the preference for maximum gap classifier biases in favor of assigning non-zero weights to redundant features (Y in the example).

On the positive side, theoretical results show that SVMs in the large sample will assign zero weights to irrelevant variables (Hardin et al., 2004). Despite this theoretical good property, in the experiments of Aliferis et al. (2010) it was found that in realistic finite sample weights of irrelevant variables are non-zero. In the work of Statnikov et al. (2006) it was found that weights of irrelevant features occasionally exceed those of weakly relevant features and furthermore that SVM weights are also susceptible to assigning larger weights to synthesis features rather than direct causes and effects.



P(T | X, Y)	(X=0, Y=0)	(X=0, Y=1)	(X=1, Y=0)	(X=1, Y=1)
T=0	1	0	0	1
T=1	0	1	1	0

P(X | A)	A=0	A=1
X=0	0.40	0.75
X=1	0.60	0.25

P(A)	
A=0	0.71
A=1	0.29

P(Y | A)	A=0	A=1
Y=0	0.80	0.10
Y=1	0.20	0.90

Figure 18: Example showing that connectivity may mitigate violations of faithfulness. Dashed line indicates a highly non-linear function (XOR). The left part shows the causal structure, while the right part shows its parameterization.

7.4. Wrapping

One of the widely-cited advantages of wrapping as a feature selection method is that it allows to tailor the selection of features to the inductive bias of the classifier (Kohavi and John, 1997). We show here how this property when combined with rich connectivity may yield causally misleading results. Consider the generative process of Figure 17. The target T is a quadratic function of its true causes A, B. Variables X, Y are effects of A, B respectively with similar non-linear functional relationships. A causal discovery procedure such as HITON-PC given enough sample and a suitable statistical test of independence will discover $\{A, B\}$ as the correct set of direct causes and direct effects. Consider however a practitioner who attacks the problem of learning a good classifier for T and reducing the necessary feature set using wrapping instead. If, as would normally be the case, the analyst starts with a simpler model class before proceeding to consider more complex ones, assuming that noise components e2, and e3 are small enough then the linear classifier would perform very well with $\{X, Y\}$ as predictors and a wrapper tailored to the linear inductive bias would eliminate A and B.

In small networks with a few variables and limited connectivity the above possibility is small, however in large networks with thousands of variables and rich connectivity as well as with massive information redundancy (e.g., biological networks) such "variable replacement" is entirely feasible and thus tailoring feature selection to a classifier's inductive bias (as wrapping does) can be an obstacle to sound causal discovery.

7.5. Connectivity and Priors Compensating for Violations of Faithfulness - Learning XOR Parents Using Univariate Association in GLL and Other Algorithms

A violation of faithfulness where constraint-based algorithms are expected to fail is when the target is an extremely non-linear function of its parents. A prototypical example is when T is the parity (XOR) of its parents A and B. Conventional wisdom, based on the truth table of the XOR function, dictates that first-order effects are zero and, as a result, the parents cannot be detected by the inclusion heuristic of the algorithm (i.e., HITON-PC or MMPC). As shown in Figure 18 however, connectivity among variables

can mitigate this difficulty. In the figure, variables X and Y can have non-zero univariate association with T, even though in textbook descriptions of parity where parents are unconnected and with 50% prior probability each for being 0 or 1, univariate association vanishes. An example parameterization that allows for this effect is given in the figure as well. This counter-intuitive phenomenon occurs because when X and Y are common effects of A, knowing the value of X is informative about A and thus about Y. Therefore the joint values of $\{X, Y\}$ are constrained and this creates univariate association of X and Y with T. Similarly, conditional association of X with T given Y is non zero. The phenomenon is not restricted to parity (or other extremely non-linear) functions in which the parity parents are connected in the network. Figure 20 in the Appendix shows an example where skewed priors on the unconnected parity parents X, Y lead to non-zero univariate association of X and Y with the target T.

The phenomenon described in this sub-section does not only apply to GLL algorithms but extends to other feature selectors as well. For example, the success of univariate filtering as feature selector, which has been documented in many domains (Guyon et al., 2006), can in part be explained via connectivity effects that allow univariate association to detect complex non-linear relationships of selected features with the target variable.

The discussion in this section is complemented by analysis of embedded feature selection in decision tree induction and of RELIEF in the online supplement Figures S5 and S6 (omitted here due to space limitations). It is shown that these algorithms can admit false positives and false negatives both predictively and causally with respect to the target variable neighborhood.

8. Discussion and Open Problems

In this section we present a thorough discussion of results, outline open problems and future directions, and provide a conclusion.

8.1. Discussion of Results

The algorithms presented, and their applied evaluation and theoretical analysis clarify many of the initially open questions discussed in Aliferis et al. (2010) and point to several new research directions. We showed that in empirical tests with 9 simulated data sets, GLL convergence to optimal performance is very fast with respect to sample size both in the sense of producing feature sets that have equal predictivity as the true $MB(T)$ and $PC(T)$ sets, and in the sense of achieving near optimal predictivity even at moderate samples sizes. These results corroborate the empirically good performance of GLL instantiations in real data sets (Aliferis et al., 2010).

An unexpected and important finding was that *GLL algorithms exhibit strong intrinsic control of false positives due not only to weakly relevant but also due to irrelevant features.* This control is empirically better in the tested data sets than what formal state-of-the-art FDR control provides except in the rare case when the data consists exclusively of irrelevant features. In Statnikov et al. (2010) we show that GLL can discover differentially expressed genes when the sample size is so small that FDR does not yield any gene. The same cannot be said for other feature selection methods that were found to be particularly prone to false positives due to both irrelevant and weakly relevant features. On the other hand, it needs to be noted that classical FDR methods do not control at all weakly relevant false positives (as GLL does). A simple pre-filtering of GLL algorithms

with an FDR control method eliminates false positives in all cases tested and yields the best algorithm for local causal learning among tested algorithms. We expect that other algorithms for example PC and MMHC will benefit from such an FDR prefiltering as well.

Within the GLL framework both the *max-k* and *h-ps* parameters control the false positives and false negatives tradeoff, through control of combined power and combined significance levels. We examined via targeted experiments and theoretical discussion the complex determination of quality of statistical decisions in GLL algorithms (aspects of which are shared by previous global constraint-based algorithms). Having two parameters to control quality of statistical decisions confers advantages since they can regulate different aspects of such decisions, and trade-off statistical quality with computational complexity.

Our efforts to explain the good predictive performance of the estimated $PC(T)$ set compared to the estimated $MB(T)$ set focused on producing explanations consistent with sufficient assumptions for Markov blanket optimality so that the good performance of the $PC(T)$ set would not be wrongly construed as entailing rejection of the theoretical assumptions, or as inability to infer the correct $MB(T)$ when the assumptions hold in the data. This is because both the results of our simulated experiments in Aliferis et al. (2010) as well as previously published experiments (Tsamardinos et al., 2003b) show that GLL algorithms estimate very well the $MB(T)$ and $PC(T)$ sets.

We also used a causal graph point of view and Markov blanket concepts to understand a variety of non-causal feature selection algorithms. This approach *provides a cohesive and fresh perspective into the behavior of several algorithms for feature selection.* We made this point by showing that the theory readily reveals why prominent feature selection methods exhibit many false positives and why they cannot be used for sound causal discovery. This complements the findings of Aliferis et al. (2010) that demonstrate empirical feature selection and causal discovery suboptimality for many state-of-the-art non-causal feature selection methods.

We discussed in detail a fundamental statistical weakness of wrapping, namely that it is prone to errors due to imperfect error estimation. This is especially the case when sample size is small whereby practical unbiased error estimators have large variance. The same problem applies implicitly to widely-used feature selection approaches such as ranking by univariate association and selecting the first k features. We showed why GLL algorithms are less sensitive to this shortcoming. In general our results show that GLL instantiations are robust enough to apply across a wide variety of domains.

Established feature selection criteria in statistics such as the AIC (Akaike Information Criterion) bare some resemblance to Markov blanket feature selection in the sense that AIC does not require classification error estimation. Specifically, AIC balances the number of features (parameters) with the likelihood of the data given a model: $\text{AIC} = 2k - 2log(L)$, where k is the number of parameters and L is the likelihood function. Model selection is driven by optimizing AIC. A critical difference however is that Markov blanket induction does not require a generative model of the data to be calculated (but relies on conditional independence tests). Given that inducing a generative model is in general harder than finding features that cannot be rendered independent of the target, and given that many recent powerful classifiers do not build generative models (e.g., SVMs) it follows that the Markov blanket induction approach has a corresponding advantage over AIC. Markov blanket induction is less model-dependent than AIC for the same reason. Note that similarly the GLL algorithms by not attempting to induce edge directionality (a task harder than edge detection,

Ramsey et al., 2006) except when absolutely necessary they avoid incurring errors in edge detection produced by false conclusions about directionality (since one type of discovery affects the other). As a result, Markov blanket induction via the GLL framework has advantages over eliciting Markov blankets by using methods that require global or local orientation.

The extensive evaluation of GLL algorithms in Aliferis et al. (2010) shows that the sufficient conditions stated in the proofs for correctness are likely to hold often, or that violations may be small. In some cases we showed that the algorithms may not fail when the assumptions are violated. Due to the critical role of non-faithfulness as a major source of possible failure we discuss it here in more detail. Faithfulness is violated in a variety of situations (Spirtes et al., 2000), notably in practice when (a) extremely non-linear or deterministic functions exist, when (b) causality cannot be localized, and when (c) variables share the same information for a response (target variable). Practical examples, respectively, are extreme epistasis in genetics, non-local causation in quantum mechanics, and gene-phenotype information redundancy in gene expression microarrays. For many additional reasons see Spirtes et al. (2000) and Meek (1995).

However, we showed that even in prototypical non-faithful functions such as XOR, the existence of unbalanced priors or the existence of connectivity among XOR parent variables of the target can make such parent variables visible again to the GLL algorithms as well as other feature selectors (e.g., univariate association filtering). We believe that this finding may have broad implications of which we mention a few. First, it explains in part the success of univariate feature selection methods in many domains since univariate filtering can pick up features that are involved in extremely non-linear functions. Second, other algorithms that are typically thought to not be able to learn such functions, such as Genetic Algorithms (Sharpe, 2000) in many situations may be able to do just that. In addition, to the extent that biological systems have evolved by evolutionary processes similar to genetic algorithms, truly extreme epistatic functions may not be as rare as previously thought. Recent proposals that suggest that such functions (i.e., biological systems) can be learned (i.e., evolved) by GAs (i.e., by evolution) through multiple objective optimization may be too pessimistic (Lenski et al., 2003). Third, previous postulates that randomized experiments (e.g., in biology, medicine and psychology) because they examine one causal factor at a time are thus unable to detect parity-like functions, may also be pessimistic (Aliferis and Cooper, 1998).

Returning to non-local causality, we point out that cognitively it is advantageous to modularize causal knowledge in order to reduce the connectivity of causal graphs and thus to control learning complexity (as well as to increase ability to store and process such knowledge with limited cognitive resources). We may thus be facing in both natural as well as artificial systems a selection bias (relative to all possible theoretical distributions) where causal systems and models of those are highly modular because it is easier to create and handle such systems and their models. Indeed in most known macroscopic causal processes (e.g., biological pathways, medicine, engineering, economics, social networks) causal systems are highly modular and thus local.

For all of the above reasons faithfulness is a very reasonable a priori, and powerful in practice, distributional assumption. At the same time at least some violations can be tolerated well by causal algorithms that are designed to use it and existing research addresses violations systematically, for example extensions of standard causal discovery algorithms capable of addressing target information equivalency (Statnikov, 2008).

The exploration of parallel and distributed techniques in the present paper showed that *GLL is amenable to parallelized and distributed local causal discovery and feature selection.* We established empirically the potential of parallelization for speeding up processing time without loss of quality. The presented parallel algorithm can also be used for distributed feature selection and causal discovery in a principled manner. Many more algorithms (namely that induce Markov blankets and admit symmetry correction when needed) can be constructed following the approach introduced in parallel and distributed IAMB for Markov blanket induction (Aliferis et al., 2002). In contrast to parallel IAMB however, parallel GLL-PC can be exponentially faster (or slower) than induction in the full data. This is a very interesting future research direction.

In exploring the transition from local-to-global strategies we showed that the local-to-global learning framework LGL can be instantiated in several ways. We examined one new instantiation of local-to-global learning, algorithm HHC. Although in most real data tested a random variable order is as good as perfectly-informed ordering by local connectivity, we showed in the present paper something previously unnoticed, namely that in some cases the right order of local neighborhood learning can entail exponential time vs. low-order polynomial time execution of local-to-global algorithms. This finding has a subtle implication: if the right ordering can be found for local learning, the resulting global learning of all variables can be faster than the local learning targeted at just one variable. Thus, just as local learning can speed up global learning the reverse may also be true.

On the other hand, our results showed that the idea that non-causal feature selection methods could help in addressing scalability of formal causal algorithms may be misplaced in light of the failure of non-causal feature selection methods to induce causality and given that highly scalable and sound methods such as GLL algorithms do exist. Several tested algorithms where non-causal feature selection is used to elicit a skeleton which is then oriented and refined by formal causal global methods are very slow and typically produce lower-quality graphs than LGL instantiations relying on sound local causal methods.

8.2. Open Problems and Future Directions

The results presented in Aliferis et al. (2010) and in the present paper merely scratch the surface of causal feature selection algorithms, local causal learning, and local-to-global learning. We briefly discuss here a few salient opportunities for moving this exciting area forward.

An assumption that is probably too strong for soundness of $MB(T)$ induction is that of causal sufficiency. For example, we conjecture without formal proof, that the algorithms should attain soundness even if the causal sufficiency is localized among the target and the members of its Markov blanket. Even when this local causal sufficiency is violated, predictive optimality among measured variables may not be compromised in many practical situations (although the usual causal interpretation of the found features is affected). Characterizing localized versions of faithfulness and causal sufficiency is an area that is likely to give a better understanding of existing algorithms and possibly lead to improvements. Examining and dealing with the effects of temporal aggregation, sampling (e.g., cellular) aggregation, feedback loops, and limited local causality on feasibility of local causal discovery will be helpful in determining the space of practical usefulness of the GLL framework.

A previously underemphasized important parameter for false negatives control is the order of conditional independence tests used for elimination (i.e., part of the elimination strategy in the GLL-PC schema). In general, the earlier time that strongly relevant variables are being examined for elimination, the better the chances for avoiding a false negative conditional independence test result since the combined power is larger. This is accomplished implicitly in HITON-PC and MMHC by using heuristics that include strongly relevant features first in $TPC(T)$ and then in both semi-interleaved HITON-PC and MMHC, where new candidates are considered for elimination *first* and where conditioning sets are constructed with stronger candidates for $PC(T)$ *first*. Systematic study of such prioritization schemes may yield performance benefits over existing GLL instantiations. Other areas that may yield improved performance is selective or full model averaging to address instability of $MB(T)$ estimation in small samples and optimizing alpha thresholds and FDR thresholds either for a domain or a data set, possibly separately for each variable.

In general, the treatment of determination of unreliable tests by means of the heuristic rule and parameter *h-ps* in GLL instantiations can be improved by incorporating formal power-size analysis whenever possible. More broadly, removing the requirement for a uniform sample size requirement across independence tests of same order (but different response function) is likely to yield improved algorithms. Other statistical issues such as improved statistical handling of structural zeros for discrete statistics, improved statistical tests that combine discrete and continuous data, handling "forced" covariates (i.e., variables that need to remain in $TPC(T)$ or $TMB(T)$ so that a particular effect is controlled for) are also worth exploring. Related to proper statistical testing is the issue of optimal discretization, not for classification as has been explored before in the literature, but for causal discovery (for a study toward that direction see Fu 2005). Other statistical extensions are to adapt the GLL method for survival analysis, or other time-to-event analyses without discretizing outcomes and with ability to handle observation censoring.

Exploitation of prior knowledge and development of methods to exploit prior causal knowledge (e.g., variable ordering, forced edges, forbidden edges, known size of local neighborhoods, known directionalities/structure and degree of connectivity, etc.) may yield greatly improved methods. Comparisons of knowledge-enhanced to pure data-driven instantiations will then be very informative.

An obvious possibility not examined in the present work is using GLL methods for regression. Another natural line of future research is to study situations where a loss function does not require exact knowledge of the conditional probability $P(T \mid MB(T))$ in which a promising strategy is to use a wrapping post-processing step to remove unnecessary features thus tailoring the final feature set to a loss function less stringent than the ones that typically guarantee soundness for GLL-MB algorithms.

Different distributional assumptions, for example monotone DAG faithfulness to make GLL and LGL algorithms faster (for a first attempt see Brown et al. 2005) may provide algorithms that tradeoff well quality for speed in specific domains.

Although we did not address the issue in this work, post-processing the results of GLL and LGL output using algorithms that detect hidden variables and orient edges is an obvious direction for research.

The study of convergence behavior of GLL and of false discovery rate control were either empirical or qualitative in the present paper. Derivation of mathematical analyses of convergence to the optimal $MB(T)$ and optimal classifier (as function of sample size), of effects of synthesis, of how common synthesis is, of combined power and alpha for

specific distributions will be very interesting, especially as other components of the framework (for example handling of unreliable tests) are also formalized.

Developing methods that handle efficiently very large neighborhoods with hundreds of features and small sample size, as well as developing methods for special-purpose causal structures (e.g., genome-wide association studies) is also an area where significant improvements can be made.

The skeleton phase of LGL is a form of dynamic programming and this explains its efficiency and soundness and probably leaves reduced opportunity for dramatic efficiency improvements. One possible avenue would be the exploration of different strategies for linking together the local skeleton results (step #2 in LGL schema). Both MMHC and HHC use an "OR" strategy but many alternative approaches can be devised. Furthermore, the edge orientation step may be greatly improved over the use of greedy search-and-score. Numerous other obvious instantiations of LGL (for instance combining GLL-PC versions with global algorithms such as GES, and TPDA) can also be implemented with substantial potential for good empirical performance. Moreover, methods to automatically identify optimal variable prioritization for local learning can yield improvements in certain distributions and we outlined related research directions in Section 6.3.

Finally, extending the framework to address broader definitions of feature selection is particularly important. Examples include finding: all sets that give desired trade-off between feature number and predictivity; all sets with smallest cost that give highest predictivity (i.e., when different observation costs apply for each variable); and all sets that optimize arbitrary multi-attribute utility/loss functions.

8.3. Conclusions

The empirical and theoretical results presented in the present paper and its companion paper (Aliferis et al., 2010) support the notion that local causal learning in the form of Markov blanket and local neighborhood induction is a theoretically well-motivated and empirically robust learning methodology as embodied in the Generalized Local Learning framework. Generalized Local Learning yields algorithms with excellent performance in data analysis geared toward classification and causal discovery. Local-to-global learning strategies have the potential to enhance large-scale causal discovery. Several existing open problems offer possibilities for non-trivial theoretical and practical discoveries, making this an exciting field of research.

Appendix A.

This Appendix provides additional tables and figures referenced in the paper.

References

C. F. Aliferis and G. F. Cooper. Aspects of modeling with mtbn's. *Technical report CBMI 1998-3, Center for Biomedical Informatics, University of Pittsburgh*, 1998.

C. F. Aliferis and A. Statnikov. Dynamic ordering-based global learning. *Technical report DSL-08-02*, 2008.

Table 15: Simulated and resimulated data sets used for experiments. The *Lung_Cancer* network is resimulated from human lung cancer gene expression data (Bhattacharjee et al., 2001) using the SCA algorithm (Friedman et al., 1999). The *Gene* network is resimulated from yeast cell cycle gene expression data (Spellman et al., 1998) using SCA algorithm. More details about data sets are provided in Tsamardinos et al. (2006).

Bayesian network	Number of variables	Training samples	Number of selected targets
Child10	200	5 x 200, 5 x 500, 1 x 5000	10
Insurance10	270	5 x 200, 5 x 500, 1 x 5000	10
Alarm10	370	5 x 200, 5 x 500, 1 x 5000	10
Hailfinder10	560	5 x 200, 5 x 500, 1 x 5000	10
Munin	189	5 x 500, 1 x 5000	6
Pigs	441	5 x 200, 5 x 500, 1 x 5000	10
Link	724	5 x 200, 5 x 500, 1 x 5000	10
Lung_Cancer	800	5 x 200, 5 x 500, 1 x 5000	11
Gene	801	5 x 200, 5 x 500, 1 x 5000	11

Table 16: Algorithms used in local causal discovery experiments with simulated and resimulated data.

HITON-PC (max k=4)	**Interleaved MMPC** (max k=2)
HITON-PC (max k=3)	**Interleaved MMPC** (max k=1)
HITON-PC (max k=2)	**HITON-MB** (max k=3)
HITON-PC (max k=1)	**MMMB** (max k=3)
Interleaved HITON-PC (max k=4)	**RFE** (reduction of features by 50%)
Interleaved HITON-PC (max k=3)	**RFE** (reduction of features by 20%)
Interleaved HITON-PC (max k=2)	**UAF-KruskalWallis-SVM** (50%)
Interleaved HITON-PC (max k=1)	**UAF-KruskalWallis-SVM** (20%)
MMPC (max k=4)	**UAF-Signal2Noise-SVM** (50%)
MMPC (max k=3)	**UAF-Signal2Noise-SVM** (20%)
MMPC (max k=2)	**L0**
MMPC (max k=1)	**LARS-EN** (for multiclass response)
Interleaved MMPC (max k=4)	**LARS-EN** (one-versus-rest)
Interleaved MMPC (max k=3)	

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using markov blanket induction for the prediction of protein-drug binding. *Technical Report DSL 02-06*, 2002.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification. part i: Algorithms and empirical evaluation. *Journal of Machine Learning*

P(T | C, D)	(D=0, C=0)	(D=0, C=1)	(D=1, C=0)	(D=1, C=1)
T=0	0.55	0.45	0.48	0.45
T=1	0.45	0.55	0.52	0.55

P(E | T, F)	(T=0, F=0)	(T=0, F=1)	(T=1, F=0)	(T=1, F=1)
E=0	0.6	0.4	0.55	0.55
E=1	0.4	0.6	0.45	0.45

P(A | B, E)	(B=0, E=0)	(B=0, E=1)	(B=1, E=0)	(B=1, E=1)
A=0	0.90	0.03	0.04	0.03
A=1	0.03	0.90	0.03	0.03
A=2	0.03	0.04	0.90	0.04
A=3	0.04	0.03	0.03	0.90

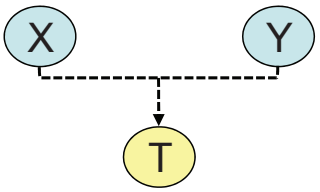
P(C)	
C=0	0.50
C=1	0.50

P(F)	
F=0	0.50
F=1	0.50

P(D)	
F=0	0.50
F=1	0.50

P(B | C)	C=0	C=1
B=0	0.98	0.02
B=1	0.02	0.98

Figure 19: Parameterization of the network in Figure 13.



P(T | X, Y)	(X=0, Y=0)	(X=0, Y=1)	(X=1, Y=0)	(X=1, Y=1)
T=0	1	0	0	1
T=1	0	1	1	0

P(X)	
X=0	0.20
X=1	0.80

P(Y)	
Y=0	0.90
Y=1	0.10

Figure 20: In this example, $T = XOR(X, Y)$. The priors of X and Y are given in the table. Both X and Y have very strong univariate association with T despite being XOR parents and in the absence of connectivity.

Research, 11:171–234, 2010.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society.Series B (Methodological)*, 57(1):289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann.Statist*, 29(4):1165–1188, 2001.

A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc.Natl.Acad.Sci.U.S.A*, 98(24):13790–13795, Nov 2001.

L. E. Brown, I. Tsamardinos, and C. F. Aliferis. A comparison of novel and state-of-the-art polynomial bayesian network learning algorithms. *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

G. Casella and R. L. Berger. *Statistical Inference*. Thomson Learning, Australia, 2nd edition, 2002.

N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: the "sparse candidate" algorithm. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

L. D. Fu. A comparison of state-of-the-art algorithms for learning bayesian network structure from continuous data. Master's thesis, Vanderbilt University, 2005.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications*. Springer-Verlag, Berlin, 2006.

D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear svm-based feature selection. *Proceedings of the Twenty First International Conference on Machine Learning (ICML)*, 2004.

I. S. Kohane, A. T. Kho, and A. J. Butte. *Microarrays for an Integrative Genomics*. MIT Press, Cambridge, Mass, 2003.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423(6936):139–144, May 2003.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. *Advances in Neural Information Processing Systems*, 12:505–511, 1999.

C. Meek. Strong completeness and faithfulness in bayesian networks. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–418, 1995.

J. Ramsey, J. Zhang, and P. Spirtes. Adjacency-faithfulness and conservative causal inference. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 2006.

M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l1-regularization paths. *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*, 2007.

O. J. Sharpe. *Towards a Rational Methodology for Using Evolutionary Search Algorithms*. PhD thesis, University of Sussex, 2000.

P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol.Biol Cell*, 9(12):3273–3297, Dec 1998.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*, volume 2nd. MIT Press, Cambridge, Mass, 2000.

A. Statnikov. Algorithms for discovery of multiple markov boundaries: Application to the molecular signature multiplicity problem. *Ph.D.Thesis, Department of Biomedical Informatics, Vanderbilt University*, 2008.

A. Statnikov, D. Hardin, and C. F. Aliferis. Using svm weight-based methods to identify causally relevant and non-causally relevant variables. *Proceedings of the NIPS 2006 Workshop on Causality and Feature Selection*, 2006.

A. Statnikov, J. Feig, E. Fisher, and C.F. Aliferis. Novel bioinformatics methods for discovery of complex molecular signatures, pathways, and biomarkers in very small sample situations. *Submitted*, 2010.

I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AI & Stats)*, 2003.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 376–381, 2003a.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 673–678, 2003b.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.