CAUSAL DISCOVERY OF DYNAMIC SYSTEMS

by

Mark Voortman

B.S., Delft University of Technology, 2005

M.S., Delft University of Technology, 2005

Submitted to the Graduate Faculty of

the School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Mark Voortman

It was defended on

December 3, 2009

and approved by

Marek J. Druzdzel, School of Information Sciences

Roger Flynn, School of Information Sciences

Stephen Hirtle, School of Information Sciences

Clark Glymour, Carnegie Mellon University

Denver Dash, Intel Labs Pittsburgh

Dissertation Director: Marek J. Druzdzel, School of Information Sciences

CAUSAL DISCOVERY OF DYNAMIC SYSTEMS

Mark Voortman, PhD

University of Pittsburgh, 2009

Recently, several philosophical and computational approaches to causality have used an interventionist framework to clarify the concept of causality [Spirtes et al., 2000, Pearl, 2000, Woodward, 2005]. The characteristic feature of the interventionist approach is that causal models are potentially useful in predicting the effects of manipulations. One of the main motivations of such an undertaking comes from humans, who seem to create sophisticated mental causal models that they use to achieve their goals by manipulating the world.

Several algorithms have been developed to learn static causal models from data that can be used to predict the effects of interventions [e.g., Spirtes et al., 2000]. However, Dash [2003, 2005] argued that when such equilibrium models do not satisfy what he calls the *Equilibration-Manipulation Commutability (EMC)* condition, causal reasoning with these models will be incorrect, making dynamic models indispensable. It is shown that existing approaches to learning dynamic models [e.g., Granger, 1969, Swanson and Granger, 1997] are unsatisfactory, because they do not perform a necessary search for hidden variables.

The main contribution of this dissertation is, to the best of my knowledge, the first provably correct learning algorithm that discovers dynamic causal models from data, which can then be used for causal reasoning even if the EMC condition is violated. The representation that is used for dynamic causal models is called *Difference-Based Causal Models (DBCMs)* and is based on Iwasaki and Simon [1994]. A comparison will be made to other approaches and the algorithm, called DBCM Learner, is empirically tested by learning physical systems from artificially generated data. The approach is also used to gain insights into the intricate workings of the brain by learning DBCMs from EEG data and MEG data.

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

This dissertation is the final product of a little over fours years of work done in the Decisions Systems Laboratory (DSL) at the University of Pittsburgh. I would like to use this section to thank several people that have been important to me over these years.

First, and foremost, I would like to thank my advisor Marek Druzdzel, for all his help and feedback during my stay at DSL. In fact, it was him who convinced me to pursue a Ph.D. in the first place. From him I learned all skills a good researcher has to possess, from finding a good research idea to writing a paper and presenting it. His advice, not only confined to research, has been invaluable. Besides Marek, I am also grateful to the other members in my Ph.D. committee. I met Roger at our weekly meetings in DSL and I always liked how he keeps asking questions to truly understand things. My cooperation with Stephen was short but pleasant. I would like to thank Clark for inviting me to a seminar given by him at CMU that was directly related to my research, and for the insightful feedback on drafts of my thesis. I actually met Denver for the second time at the seminar at CMU and he motivated me to continue the work where he left off. He was always quick to point out any mistakes in my reasoning and his feedback has been incredibly helpful in developing my ideas. Our collaboration has resulted in several papers and I hope more will follow in the future.

I am also grateful to all the people that surrounded me in the School of Information Sciences. The staff were always very helpful in answering any questions I had. I would also like to thank all the people, too many to list here, that had to put up with me in DSL. I always felt at home and many of my colleagues eventually turned into friends.

Last, but not least, I would like to thank my family and friends who were always supportive and provided much needed distractions from work. I would like to especially thank my parents, Kees and Hennie Voortman, for their love and support throughout my life.

1.0 INTRODUCTION

Recently, several philosophical and computational approaches to causality have used an interventionist framework to clarify the concept of causality [Spirtes et al., 2000, Pearl, 2000, Woodward, 2005]. The characteristic feature of the interventionist approach is that causal models are potentially useful in predicting the effects of manipulations. One of the main motivations of such an undertaking comes from humans, who seem to create sophisticated mental causal models that they use to achieve their goals by manipulating the world.

Woodward [2005] presents an elaborate interventionist account of causality by circumventing known problems in previous interventionist approaches. Those approaches tended to be anthropological and manipulations were motivated from that viewpoint only. Woodward points out, rightfully, that it is not about manipulations that can currently be performed by humans, but manipulations that can be performed potentially. Another common objection is circularity. For example, consider a causal relationship between two variables X and Y, where X causes Y, for which I will also use the notation $X \rightarrow Y$. This causal relationship presumably exists if manipulating variable X results in a change in variable Y. However, in order to define a manipulation, it is necessary to use the concept of causality because the manipulation causes a change in X, resulting in circularity. Woodward argues that in this situation really two causal relationships are under consideration, namely one between X and Y, and one involving the manipulation of X. So in order to characterize the causal relationship between X and Y, if any, we do not presume any causal information about this relationship, thereby removing the circularity. With these objections to the interventionist approach out of the way, Woodward then continues by detailing his philosophical undertaking by using the frameworks of Spirtes et al. [2000] and Pearl [2000].

The work of Spirtes et al. [2000] and Pearl [2000] focuses mainly on the computational

and algorithmic aspects. They developed most of their ideas on causality in the late eighties and early nineties of the last century. In essence, their work is equivalent although they use different terminology. I will use the terminology and framework of Spirtes et al. [2000] in this dissertation.

The main importance of the approaches mentioned in the previous paragraph was that they made it feasible to learn causal relationships from data by satisfying only a few basic assumptions. The main theorem, sometimes called the causal discovery theorem, makes it possible to direct edges in an adjacency graph if, so called, unshielded colliders are present. An unshielded collider is a subset of a graph such that three nodes, say X, Y and Z, form a causal structure $X \rightarrow Y \leftarrow Z$, and there is no edge between X and Z. The reason these unshielded colliders can be used for causal discovery is that they imply a unique independence fact, namely that X is unconditionally independent of Z. More details will be presented in later chapters and Appendix A. This simple, but powerful, discovery is slowly changing the landscape in machine learning and statistics, areas usually only focusing on correlations and not necessarily causation and interventions or manipulations.

1.1 PROBLEM STATEMENT AND MOTIVATION

Several algorithms have been developed to learn causal models from data. These models have been postulated to be useful in predicting the effects of interventions [Spirtes et al., 2000, Pearl, 2000]. However, Dash [2003, 2005] argued that when equilibrium models do not satisfy the *Equilibration-Manipulation Commutability* (EMC, for short) condition, causal reasoning with these models will be incorrect. The EMC condition is illustrated in Figure 1. Suppose we have a certain dynamic system S and we want to perform causal reasoning on that system. Many existing approaches first wait for the system to equilibrate and then collect data to learn a causal model. Finally, the learned causal model is used to predict the effect of manipulations. This approach is illustrated by path A. Alternatively, if one is able to learn the dynamic model directly from time series data, one could perform the manipulation on the dynamic graph and then equilibrate the graph. This is illustrated by path B. If both

paths lead to the same prediction, the EMC condition is satisfied. Dash [2003, 2005] showed that the EMC property is not always satisfied, and gave sufficient conditions to both obey and to violate the EMC property. Examples of both cases will be given in a later chapter. Intuitively, the reason why the EMC condition is not always satisfied is that a manipulation will move a system out of equilibrium back into a dynamic state. It is not obvious that when the system reaches equilibrium again, the causal structure is the same as before (except for the manipulation) and, as Dash showed, that is indeed not always the case.

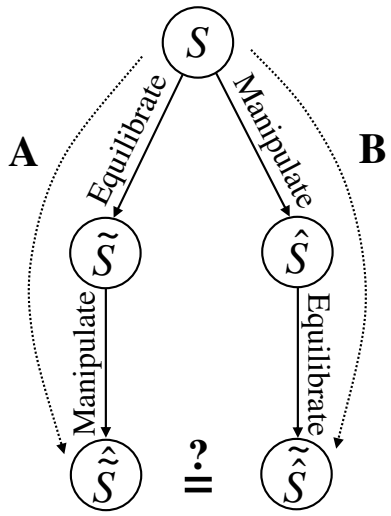

Figure 1: The EMC property is satisfied if and only if path A leads to the same prediction as path B.

Currently, no algorithms exist that are able to follow path B. There are existing approaches that learn dynamic models Granger [e.g., 1969], Swanson and Granger [e.g., 1997], but it will be explained later that because these approaches do not identify hidden variables, they lead to infinite-order Markov models that are unsuitable for causal inference. In this dissertation, an algorithm will be presented for learning dynamic models that can be used for causal inference even if EMC is violated. To the best of my knowledge, this line of research has not been pursued until now and so the main focus of this dissertation will be on developing a representation for dynamic models and learning them from data.

1.2 CONTRIBUTIONS

The main contribution of this dissertation is a provably correct learning algorithm, the DBCM Learner, that can discover dynamic causal models from data. To the best of my knowledge, this is the first algorithm that does not suffer from the problem associated with the EMC condition, because, under certain assumptions, manipulations can be performed on the dynamic graph. It is shown that existing approaches to learning dynamic models [e.g., Granger, 1969, Swanson and Granger, 1997] are unsatisfactory, because they do not perform a necessary search for hidden variables. As far as I know, it is also the first algorithm that is able to learn causal models from data generated by physical systems, such as a coupled harmonic oscillator.

The representation for dynamic causal models that is developed in this dissertation will be called *difference-based causal models (DBCMs)* and is based on Iwasaki and Simon [1994]. They represent systems of difference (or differential) equations that are used to model many real-world systems. The Iwasaki-Simon representation suffers from several limitations, such as not being able to include higher order derivatives directly and having unnecessary definitional links. My proposed representation, difference-based causal models, will remove these limitations and form a coherent and intuitive representation of any dynamic model that can be represented as a set of difference (or differential) equations.

A comparison will be made to other approaches and it is shown that the DBCM Learner uses a compact representation for physical systems that cannot be matched by other existing approaches. The reason for this is that DBCM learning searches for latent variables in the form of derivatives. This implies that models that rely on derivatives in their representation will be learned correctly by the DBCM Learner, whereas approaches that try to marginalize them out (e.g., learning vector autoregression models) will fail. I also prove that, under standard assumptions for causal discovery, the DBCM Learner will completely identify all instantaneous feedback variables, removing an important obstacle to predicting when an equilibrium version of the model will obey EMC, and allowing the model to be used to correctly predict the effects of manipulating variables.

Besides the theoretical work there is also a comprehensive evaluation of DBCM learning in

practice. The experiments can be divided into two parts. The first part focuses on relearning DBCMs from data generated from gold standards based on existing physical systems. In the second part, the DBCM Learner is used to gain insights into the intricate workings of the brain by learning models from EEG data and MEG data.

1.3 BACKGROUND

The main goal of this section is to place my line of research in a somewhat broader context. Several concepts are covered that either serve as motivation or will be of importance in later chapters.

1.3.1 Introduction to Causality

The central topic of this thesis is causality. It seems safe to say that the concept of causality, which was already discussed by Aristotle and possibly earlier, has given rise to many controversies. Hume [1739], for example, argued that causality is neither grounded in formal reasoning nor in the physical world. Therefore, he concluded, causality is nothing more or less than a habit of mind, albeit a useful one. Russell [1913] went as far as saying that causality is "a relic of bygone age," although he later retracted from this view and admitted that causality plays an important role in science.

Intuitively, causality is a relationship between two events, where the occurence of the cause will, possibly probabilistically, result in the effect. For example, it is nowadays widely accepted that smoking is a cause of lung cancer, albeit a probabilistic one. Not everyone who smokes will develop lung cancer, and not everyone who develops lung cancer smoked. However, smoking increases the chance of lung cancer. This example also shows the importance of knowledge of causal relationships as, at least in this case, it could save lives.

There is a large body of evidence that shows that humans have a disposition for learning causal relationships. Sloman [2005], for example, uses causal models to explain human decision making and shows how causal reasoning is embedded in natural language. Given

that our nervous system, and especially our brain, uses a disproportionally large amount of energy to maintain these causal models, it must be of evolutionary benefit and imperative to our survival.

In everyday life, the notion of causation is closely related to the notion of intervention, or manipulation, and for good reasons. For example, we know that starting a car by turning the key and having gas in the tank causes the car to start. It also implies that if we remove all gas from the tank (a manipulation), the car would not start. Similarly, if we know that smoking increases the chances of lung cancer, then quitting smoking (or not starting) will decrease the chances of lung cancer. The point here is that causation and manipulation are very practical concepts and intimately related to each other and, hence, worthwhile studying. This is also evident in science where the increasing amount of causal knowledge is utilized to develop, for example, more effective medicine not by just removing the symptoms but by curing the underlying causes of a disease. The more detailed knowledge one has about causal relations, the better the predictions of interventions will be. This knowledge is also very important in, for example, policy making, where decision makers have to decide what course of actions to take to obtain the highest possible benefit.

I favor an interventionist approach, such as advocated by Spirtes et al. [2000], Pearl [2000], and Woodward [2005]. These recent developments in philosophy and AI have shown that plausible versions of an interventionist approach can be developed.

1.3.2 Observations Versus Manipulations

There are two fundamental ways in which agents can interact with the world. One is via observations that are mediated by sensory inputs. The other is via manipulations, where the state of the world is changed, and, in case of humans, is executed by our bodies and directed by our brain. In other words, it is the difference between *seeing* and *doing*. It is very important to realize this distinction. Some applications, such as detection of credit card fraud, rely solely on observations, whereas other applications, such as policy making, directly involve manipulations.

Manipulations play a major role in causality and the definition of one usually mentions

the other. Humans manipulate the world all the time to find causal connections, e.g., when trying to find out why a car does not start, we turn on the light to see if the battery is dead. One limitation that humans face is that there are many manipulations that can not be performed practically, such as on the weather. Although the number of things we can manipulate increases over time, we will never be able to perform all potential manipulations. This poses a question about the limits of causal discovery, namely, if we are able to learn causal relationships from observations only and under what assumptions. This question has been answered in the affirmative by Spirtes et al. [2000] and Pearl [2000] and the assumptions have been laid out.

1.3.3 Learning Dynamic Models

The world around us is dynamic. The brain, for example, is continuously perceiving outside stimuli and reacting in appropriate ways. In causal discovery, however, the trend in the last decades was to focus on static data where the data sets simply consist of records and time plays no role. Causal learning is the act of inferring a causal model from data. The type of data serves as a constraint on what kind of models we can learn. Static data, i.e., data in which there is no time component, is used to learn static models such as Bayesian networks. In the past 20 years in AI, the practice of learning causal models from data has gained much momentum [cf., Pearl and Verma, 1991, Cooper and Herskovits, 1992, Spirtes et al., 2000]. These methods are based on the formalism of structural equation models (SEMs), which originated out of the econometrics literature over 50 years ago [cf., Strotz and Wold, 1960], and Bayesian networks [Pearl, 1988] which started the paradigm shift of graphical models in AI and machine learning 20 years ago. These methods have predominately focused on the learning of equilibrium (static) causal structure, and have recently gained inroads into mainstream scientific research, especially in biology [cf., Sachs et al., 2005].

Despite the success of these static methods, one should keep in mind that they are susceptible to the problem associated with the EMC condition. And many real-world systems are dynamic in nature and are well-modeled by systems of simultaneous differential equations. Such systems have been studied extensively in econometrics over the past four decades:

Granger causality [cf., Granger, 1969, Engle and Granger, 1987, Sims, 1980] and vector autoregression [Swanson and Granger, 1997, Demiralp and Hoover, 2003] methods have become very influential. In AI, there has been work on learning Dynamic Bayesian Networks (DBNs) [Friedman et al., 1998a] and modified Granger causality [Eichler and Didelez, 2007]. All of these structural models for dynamic systems have a very similar form. They are all discrete-time systems where there may exist arbitrary causal relations across time. While this view is general, it does not exploit the fact that several constraints are imposed on inter-temporal causal edges if the underlying dynamics are solely governed by differential equations.

In this dissertation, a new approach to learning dynamic models is presented.

1.4 NOTATION

Here is a list of notation that will be used in subsequent chapters.

- $|\boldsymbol{S}|$: denotes the number of elements in set $\boldsymbol{S}$.
- $\boldsymbol{Pa}_G(X)$: the set of parents of X in graph G. G may be omitted if it is clear from the context.
- $\boldsymbol{Ch}_G(X)$: the set of children of X in graph G. G may be omitted if it is clear from the context.
- $\boldsymbol{Anc}_G(X)$: the set of ancestors of X in G. G may be omitted if it is clear from the context.
- $\boldsymbol{Desc}_G(X)$: the set of descendents of X in G. G may be omitted if it is clear from the context.
- $\boldsymbol{NonDesc}_G(X)$: the set of non-descendents of X in G. G may be omitted if it is clear from the context.
- $\boldsymbol{Adjacencies}(G, X)$: the set of adjacencies of X in G.
- $\boldsymbol{Sepset}(X, Y)$: a set of variables $\boldsymbol{C}$, such that $(X \perp\!\!\!\perp Y \mid \boldsymbol{C})$.
- $\Delta^n V$ denotes the nth derivative of variable V, and $\Delta^0 V = V$.
- $\dot{V}$ and $\ddot{V}$ denote the first and second derivative of V, respectively.

1.5 ORGANIZATION OF THE DISSERTATION

The remainder of this dissertation consists of 3 chapters and several appendices. The next chapter will cover the theoretical aspects of DBCMs and learning them from data. Chapter 3 presents the experimental results. Chapter 4 contains a discussion. Proofs and some of the related work is presented in the appendices.

2.0 DIFFERENCE-BASED CAUSAL MODELS

This chapter introduces a representation for dynamic models called *Difference-Based Causal Models (DBCMs)*. The first section introduces the representation that is based on structural equation models. The section after that shows how to reason with DBCMs and explains the EMC condition in detail. The third section is the most important section of this chapter and treats in detail the DBCM Learner. The last few sections examine the assumptions and compare the DBCM approach to other approaches.

2.1 REPRESENTATION

In order to talk about dynamic models meaningfully, one first has to establish a representation. For this purpose, I will introduce *Difference-Based Causal Models (DBCMs)*, which are a class of discrete-time dynamic models that model all causation across time by means of difference equations driving change in the system. This representation is motivated by real-world physical systems and is derived from a representation introduced by Iwasaki and Simon [1994]. They can be seen as a restricted form of (dynamic) structural equation models that will be introduced first.

2.1.1 Structural Equation Models

Structural equation models (SEMs) are a representation that are used as a general tool for modeling static causal models and originated in the econometrics literature [cf., Strotz and Wold, 1960], but have also been discussed more recently by [Pearl, 2000], for example.

Informally speaking, a structural equation model is a set of variables $\boldsymbol{V}$ and a set of equations $\boldsymbol{E}$ in which each equation $E_i \in \boldsymbol{E}$ is written as $V_i := f_i(\boldsymbol{W}_i) + \epsilon_i$, where $V_i \in \boldsymbol{V}$, $\boldsymbol{W}_i \subset \boldsymbol{V} \setminus V_i$, and ϵ_i is a random variable that represents a noise term. Historically, especially in econometrics, SEMs use normally distributed noise terms. The equations are given a causal interpretation by assuming that the $\boldsymbol{W}_i$ are causes of V_i. The noise terms are intended to represent the set of causes of each variable that are not directly accounted for in the model.

Definition 1 (structural equation model (SEM)). *A structural equation model is a pair $\langle \boldsymbol{V}, \boldsymbol{E} \rangle$, where $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$ is a set of variables and $\boldsymbol{E} = \{E_1, E_2, \ldots, E_n\}$ is a set of equations such that each equation E_i is written as $V_i := f_i(\boldsymbol{W}_i) + \epsilon_i$, where $\boldsymbol{W}_i \subset \boldsymbol{V} \setminus V_i$ and ϵ_i is an independently distributed noise term.*

As an example, look at the following system. Let $\boldsymbol{V} = \{A, B, C, D\}$, $\boldsymbol{E} = \{E_1, E_2, E_3, E_4\}$, and let the equations be defined as follows:

$$
\begin{aligned}
E_1 &: \quad A := \epsilon_A \\
E_2 &: \quad B := \epsilon_B \\
E_3 &: \quad C := f_C(A, B) + \epsilon_C \\
E_4 &: \quad D := f_D(C) + \epsilon_D
\end{aligned}
$$

Together, these components form a structural equation model $M = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$.

A structural equation model implicitly defines a causal model by designating the variables at the right hand side to be direct causes of the variable at the left hand side of each equation, and direct effect is defined in a similar way.

Definition 2 (direct cause and effect). *Let $M = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$ be a structural equation model and $V_i := f_i(\boldsymbol{W}_i) + \epsilon_i$ an equation in this model. All $W \in \boldsymbol{W}_i$ are direct causes of V_i and V_i is a direct effect of all $W \in \boldsymbol{W}_i$. I will use $\boldsymbol{Pa}(X)$ to denote the direct causes (parents) of X and $\boldsymbol{Ch}(X)$ to denote the direct effects (children) of X.*

The set of variables in a SEM can be partitioned into a set of exogenous and a set of endogenous variables. Exogenous variables have their causes outside of the system under consideration, and endogenous variables have their causes within the system under consideration.

Definition 3 (exogenous variable)**.** *Let $M = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$ be a structural equation model. Then a variable $V_i \in \boldsymbol{V}$ is an exogenous variable relative to M if and only if it does not have any direct causes.*

Definition 4 (endogenous variable)**.** *A variable is endogenous if it is not exogenous.*

A SEM defines a directed graph such that each variable $V_i \in \boldsymbol{V}$ is represented by a node and there is an edge from each parent to its child, i.e., $V_j \to V_i$ for each $V_j \in \boldsymbol{Pa}(V_i)$. In this way, SEMs can model relations between variables in a very general way. For example, Druzdzel and Simon [1993] show that SEMs are a generalization of Bayesian networks.

Definition 5 (structural equation model graph)**.** *A graph for a structural equation model $M = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$ is constructed by directing an edge $W \to V_i$ for each $W \in \boldsymbol{W}_i$ in equation $V_i := f_i(\boldsymbol{W}_i) + \epsilon_i$, where $\boldsymbol{W}_i \subset \boldsymbol{V} \setminus V_i$ and ϵ_i is an independently distributed noise term.*

The structural equation model graph, to which I will also refer as *causal graph*, for the example is given in Figure 2.

In this dissertation I will assume that the SEM graphs are acyclic. This is a standard assumption in causal discovery. After I introduce DBCMs I will give a better justification of using acyclic graphs.

Assumption 1 (acyclicity)**.** *All structural equation model graphs are acyclic.*

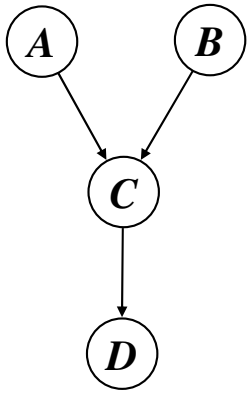

Figure 2: The causal graph for the SEM example.

2.1.2 Dynamic Structural Equation Models

Dynamic structural equation models (DSEMs) are the temporal extension of structural equation models. In this dissertation I will assume a discrete-time setting, i.e., the equations will be difference equations and not differential equations. Each variable in a DSEM can have causes in the same time slice just like a SEM, but also in previous time slices. This is made explicit in the following definition, which is slightly more complicated than the definition for SEMs.

Definition 6 (dynamic SEM)**.** *A dynamic SEM is a pair* $\langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$*, where* $\boldsymbol{V^t} = \{V_1^{t_1}, V_2^{t_2}, \ldots\}$ *is a set of time indexed variables such that* $t_i = t_j$ *for all* $i, j \leq n$*, and* $t_k < t_i$ *for all* $k > n$*.* $\boldsymbol{E} = \{E_1, E_2, \ldots, E_n\}$ *is a set of equations such that each equation* E_i *is written as* $V_i^t := f_i(\boldsymbol{W^s}_i) + \epsilon_i^t$*, where* $\boldsymbol{W^s}_i \subset \boldsymbol{V^t} \setminus V_i^{t_i}$*, and* ϵ_i^t *is an independently distributed noise term.*

Simply put, each variable in time slice i is a function of other variables in time slice i and variables from time slices before i. Note that when none of the variables in time slice i is dependent one a variable from a previous time slice, a dynamic SEM reduces to a SEM. For completeness, it is required to also define initial conditions but for simplicity this has been omitted. Here is the example from before extended to a dynamic SEM, where $\boldsymbol{V^t} = \{A^t, B^t, C^t, D^t, C^{t-1}, D^{t-1}, D^{t-2}\}$ and $\boldsymbol{E} = \{E_1, E_2, E_3, E_4\}$:

$$
\begin{aligned}
E_1: &\quad A^t := f_A(C^{t-1}) + \epsilon_A^t \\
E_2: &\quad B^t := \epsilon_B^t \\
E_3: &\quad C^t := f_C(A^t, B^t) + \epsilon_C^t \\
E_4: &\quad D^t := f_D(D^{t-2}, D^{t-1}, C^t) + \epsilon_D^t
\end{aligned}
$$

A dynamic SEM graph is defined analogous to a regular SEM graph.

Definition 7 (dynamic SEM graph)**.** *A graph for a dynamic SEM* $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ *is constructed by directing an edge* $W^s \to V_i^t$ *for each* $W^s \in \boldsymbol{W^s}_i$ *in equation* $V_i^t := f_i(\boldsymbol{W^s}_i) + \epsilon_i^t$*,*

where $\boldsymbol{W}^s{}_i \subset \boldsymbol{V}^t \setminus V_i^t$, $s \leq t$ *for every* $W^s \in \boldsymbol{W}^s{}_i$, *and* ϵ_i^t *is an independently distributed noise term.*

A dynamic SEM graph is an infinite graph and I will use two ways to draw this graph in finite space. The first way I will call the *shorthand* graph and an example is shown in Figure 3. All the nodes and edges in time slice t are shown, plus the nodes from previous time slices that are direct causes of nodes in time slice t. For convenience, I will used dashed arcs for cross-temporal relationships.

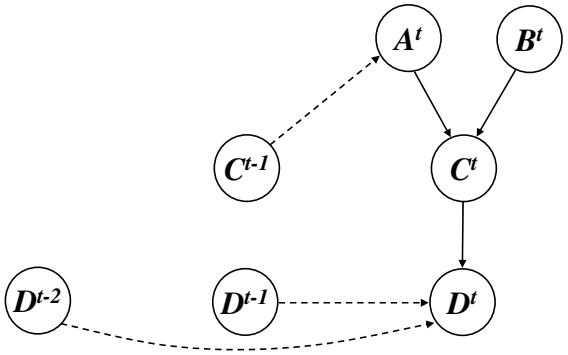

Figure 3: The shorthand causal graph for the dynamic SEM example.

The second way simply displays the graph for several time slices and I will call this the *unrolled* version. The unrolled version of four time slices of the example is shown in Figure 4. This graph also makes it clear that initial conditions are required for a fully specified model. For example, A^0, D^0, and D^1 are not specified by the equations given before and should be defined separately.

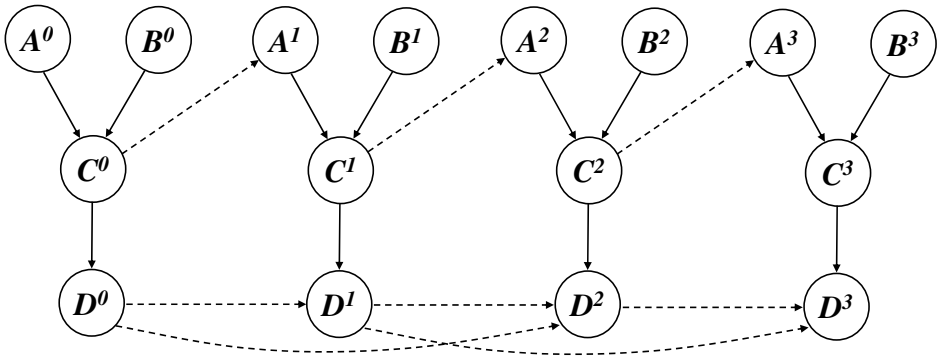

Figure 4: The unrolled causal graph for the dynamic SEM example.

2.1.3 Difference-Based Causal Models

Stated briefly, a DBCM is a set of variables and a set of equations, with each variable being specified by an equation. The defining characteristic of a DBCM is that all causation across time is due to a derivative (e.g., $\dot{x}$) causing a change in its integral (e.g., x). Equations describing this relationship are called *integral equations* (e.g., $x^t = x^{t-1} + \dot{x}^{t-1}$) and are deterministic. In addition, contemporaneous causation is allowed, where variables can be caused by other variables in the same time slice. DBCMs are dynamic structural equation models, but with the cross-temporal restriction imposed by integral equations.

Definition 8 (difference-based causal model (DBCM))**.** *A difference-based causal model* $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ *is a dynamic SEM where the only across time equations allowed are of the form* $V_i^t := V_i^{t-1} + V_j^{t-1}$*, where* $i \neq j$*.*

Definition 9 (integral variable and equation)**.** *Let* $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ *be a DBCM. Then* $V_i^t \in \boldsymbol{V^t}$ *is an integral variable if there is an equation* $E_i \in \boldsymbol{E}$ *such that* $V_i^t := V_i^{t-1} + V_j^{t-1}$*, where* $i \neq j$*.* E_i *is an integral equation.*

The only variables that have causes in a previous time slice are the integral variables (the other types of variables will be named shortly). Aside from edges into the variables determined by integral equations, no causal edges are allowed between time slices. This implies that if a variable is not an integral variable, all its causes and effects are within the same time slice. This restriction makes DBCMs a subset of causal models as they were defined in the previous section. This excludes dynamic models that have time lags greater than one, but it does still include all physical systems based on ordinary differential equations.

Here is the simple example converted into a DBCM, where A^t has become an integral variable:

$$
\begin{aligned}
E_1: &\quad A^t := A^{t-1} + D^{t-1} \\
E_2: &\quad B^t := \epsilon_B^t \\
E_3: &\quad C^t := f_C(A^t, B^t) + \epsilon_C^t \\
E_4: &\quad D^t := f_D(C^t) + \epsilon_D^t
\end{aligned}
$$

The variables in an integral equation are clearly related to each other and it is useful to have terminology to refer to their relationships.

Definition 10 (difference)**.** *Let $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ be a DBCM. Then $V_j^t \in \boldsymbol{V^t}$ is a difference of $V_i^t \in \boldsymbol{V^t}$, denoted ΔV_i^t, if there is an equation in $\boldsymbol{E}$ such that $V_i^t := V_i^{t-1} + V_j^{t-1}$, where $i \neq j$.*

Intuitively, as $\Delta t \to 0$ a difference $\frac{\Delta V_i}{\Delta t} \to \frac{dV_i}{dt}$, so I will refer to differences sometimes as *derivatives*.

A difference relationship is defined recursively and, in fact, we can speak about second and third differences as well. In general, I will use the notation $\Delta^n V$ to denote the nth derivative of V, and $\Delta^0 V = V$ by definition. Sometimes I will shorten $\Delta^1 V$ to ΔV if there are no other derivatives defined. As an alternative notation, sometimes I will use the physics notation, i.e., $\dot{V}$, to denote derivatives. Once we are accustomed to the language of differences, it is not necessary anymore to explicitly write down integral equations because they are implied, and I will usually omit them.

Again, here is the example from before but now it is using the derivative relationship, making it more intuitive:

$$
\begin{aligned}
E_1: &\quad A^t := A^{t-1} + \dot{A}^{t-1} \\
E_2: &\quad B^t := \epsilon_B^t \\
E_3: &\quad C^t := f_C(A^t, B^t) + \epsilon_C^t \\
E_4: &\quad \dot{A}^t := f_{\dot{A}}(C^t) + \epsilon_{\dot{A}}^t
\end{aligned}
$$

A DBCM graph is created in a way analogous to a graph for a dynamic SEM.

Definition 11 (DBCM graph)**.** *A graph for a DBCM* $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ *is constructed by directing an edge* $W^s \to V_i^t$ *for each* $W^s \in \boldsymbol{W^s}_i$ *in equation* $V_i^t := f_i(\boldsymbol{W^s}_i) + \epsilon_i^t$*, where* $\boldsymbol{W^s}_i \subset \boldsymbol{V}^t \setminus V_i^t$*,* $s \leq t$ *for every* $W^s \in \boldsymbol{W^s}_i$*, and* ϵ_i^t *is an independently distributed noise term.*

Just like with SEMs, I will assume that DBCMs are acyclic structures. Cyclic graphs are used to model systems with instantaneous feedback loops [Richardson and Spirtes, 1999]. In DBCMs, I assume that the sampling rate of the data is so high that the actual feedback loops can be detected, and those loops always involve integral variables.

Again, we can draw a shorthand or unrolled version of the graph. The shorthand graph is displayed at the left of Figure 5. However, because an integral variable always involves itself and its derivative, it is sufficient to draw a dashed arc from the derivative to the variable, indicating an integral relationship (this also obviates the need for time indices). It is important to note that this way cycles arise, but it really is an acyclic structure over time. This is clear if we look at the unrolled graph in Figure 6.

A DBCM can be partitioned into three types of variables, namely *integral*, *prime*, and *static* variables. Each of the type of variables is determined by an equation of the corresponding type.

Integral variables were already defined. The term *integral* usually refers to summing continuous variables; however, intuitively it gives a better feel for what is happening to these variables over time than the term *summation* variables would. In the example, variable A is an integral variable. Integral variables are determined over time by an integration chain,

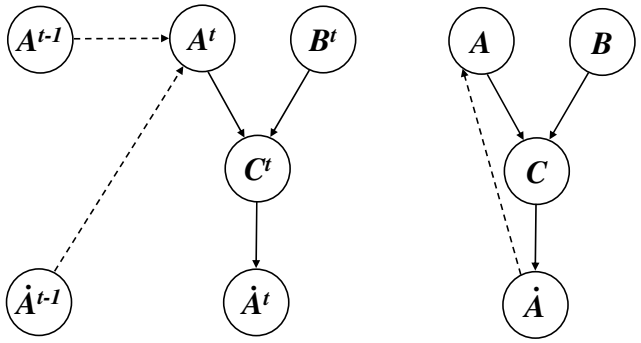

Figure 5: Left: The shorthand causal graph for the DBCM example. Right: Same shorthand graph as the left hand side, but simplified by drawing the integral relationship from the derivative ($\dot{A}$) to the integral (A), as well as dropping the time indices.

e.g., $\ddot{x} \rightarrow \dot{x} \rightarrow x$. Here, x and $\dot{x}$ are integral variables, and $\ddot{x}$ is called a *prime* variable. All the changes in integral variables are ultimately driven by prime variables.

Definition 12 (prime variable and equation)**.** *Let $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ be a DBCM. Then $V_j^t \in \boldsymbol{V^t}$ is a prime variable if it is contained in an integral equation but is not a integral variable.*

Variable $\dot{A}$ is a prime variable, because it is contained in integral equation E_1 but is not an integral variable itself. The last type of variable are static variables.

Definition 13 (static variable and equation)**.** *A variable V_j^t is a static variable if it is neither an integral variable nor a prime variable. The equation $E_i \in \boldsymbol{E}$ that has V_j^t as an effect is a static equation.*

A static variable is conceptually equal to a prime variable of zeroth order, however, it is not part of an integration chain the way prime variables are. The DBCM Learner that is described later on does not make a distinction in the way prime variable and static variables are detected. The term static variable does not imply that the variable is not changing from time-step to time-step, because it might be part of a feedback loop. However, I use this

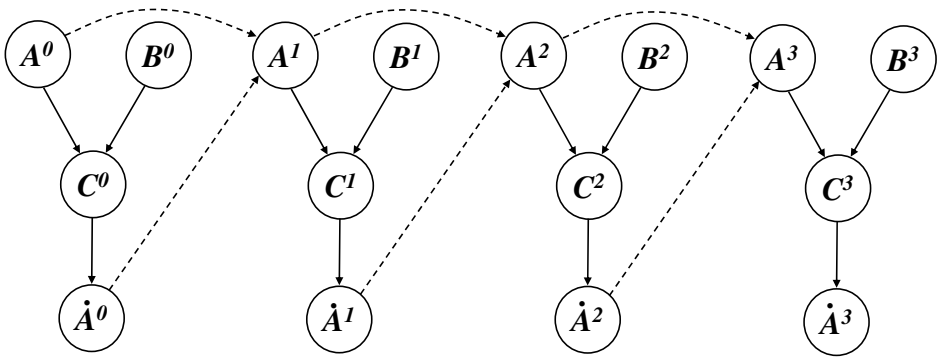

Figure 6: The unrolled causal graph for the DBCM example.

term to emphasize that their only causes are contemporaneous (and they are not part of an integration chain like prime variables). Variables B and C are static variables.

As a more concrete example, consider the set of equations describing the motion of a damped simple harmonic oscillator, shown in Figure 7. A block of mass m is suspended from a spring in a viscous fluid. The harmonic oscillator is an archetypal dynamic system, ubiquitous in nature. Abstractly, it represents a system whose "restoring force" is proportional to distance from equilibrium, and as such it can form a good approximation to many nonlinear systems close to equilibrium. Furthermore, the $F = ma$ relationship is a canonical example of causality: applying a force to cause a body to move. Thus, although this system is simple, it illustrates many important points, and can in fact be quite complicated using standard representations for causality, as I will show later.

Like all mechanical systems, the equations of motion for the harmonic oscillator are given by Newton's 2nd law describing the acceleration a of the mass under the forces (due to the weight, due to the spring, F_x, and due to viscosity, F_v) acting on the block. These forces *instantaneously* determine a; furthermore, they indirectly determine the values of all integrals of a, in particular the velocity v and the position x, of the block. The longer time passes, the more influence those forces have on the integrals. Although the simple harmonic oscillator

is a simple physical system, having noise is still quite realistic: e.g., friction, air pressure, temperature, all of these factors are weak latent causes that add noise when determining the forces of the system. Writing this continuous time system as a discrete time model leads to the following DBCM:

$$
\begin{aligned}
E_1: &\quad a := f_a(F_x, F_v, m) + \epsilon_a \\
E_2: &\quad F_x := f_{F_x}(x) + \epsilon_{F_x} \\
E_3: &\quad F_v := f_{F_v}(v) + \epsilon_{F_v} \\
E_4: &\quad m := \epsilon_m
\end{aligned}
$$

Please note that the time indices have been dropped for simplicity, as they are implicitly defined by the integral relationships between a, v, and x. In this model, variables x and v are the integral variables, for which the equations are:

$$
\begin{aligned}
v^t &:= v^{t-1} + a^{t-1} \\
x^t &:= x^{t-1} + v^{t-1}
\end{aligned}
$$

An integral variable X_i is part of a chain of causation where the (non-reflexive) parent ΔX_i of X_i is a variable that may in turn be an integral variable itself, or it may be the highest-order derivative of X_i, in which case it can have only contemporaneous parents. Variable a is the prime variable of the integration chain that involves x and v as well. Variables m, F_d and F_x in the example are static variables. The shorthand and unrolled graph for two time slices are displayed in Figure 8 and 9, respectively.

DBCM-like models were discussed in great detail by Iwasaki and Simon [1994] and Dash [2003, 2005] (see Appendix B). However, the Iwasaki-Simon representation suffers from several limitations, such as not being able to include higher order derivatives directly and having unnecessary definitional links (see Iwasaki and Simon [1994] for details). Iwasaki and Simon

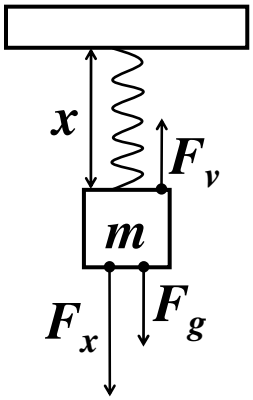

Figure 7: A simple harmonic oscillator.

only allow first order derivatives in their mixed structures, but by the fact that any higher order differential equation can be transformed into a system of first order differential equations, their approach is equally general but their graphs can be hard to interpret. My contribution to the representation is to add syntax that distinguishes variables that are determined by integral equations from those that are not, and to support higher order derivatives directly without transforming them to systems of first order derivatives by variable substitution. DBCMs remove these limitations and form a coherent representation of any dynamic model that can be represented as a set of difference (or differential) equations.

While there exist mathematical dynamic systems that can not be written as a DBCM, I believe that systems based on differential equations are ubiquitous in nature, and therefore will be well approximated by DBCMs. Furthermore, because DBCMs have much more restricted structures than arbitrary causal models over time, they can, in principle, be learned much more efficiently and accurately as we will see in a later section.

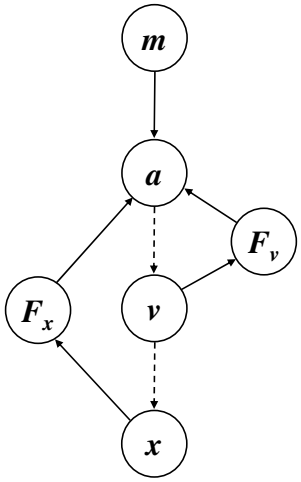

Figure 8: The shorthand causal graph of the simple harmonic oscillator.

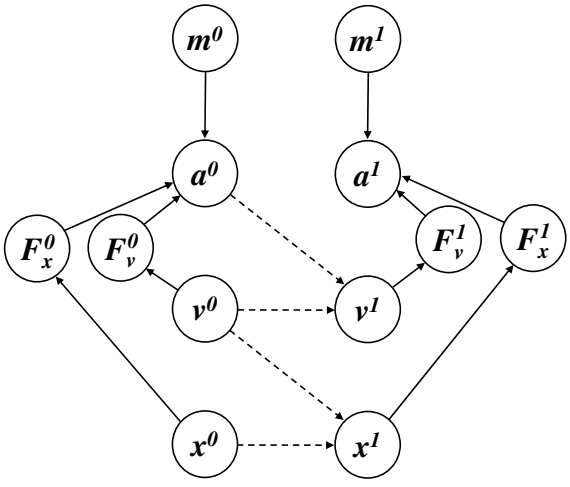

Figure 9: The unrolled causal graph of the simple harmonic oscillator.

2.2 REASONING

In this section I will discuss two important types of reasoning that can be performed with DBCMs, namely equilibration and manipulation. After introducing these types of reasoning, I present a very important caveat in reasoning with dynamic systems that was introduced in Dash [2003]. This caveat is the main motivation why it is important to learn dynamic models. This section is somewhat informal in tone to give the general ideas, for more details please consult Dash [2003].

2.2.1 Equilibrations

The first type of reasoning that will be discussed is what happens when a variable in a system reaches equilibrium. Intuitively, an equilibration is a transformation from one model into another where the derivatives of one of the variables have become zero. Equilibration is formalized by the *Equilibrate* operator. The procedure is similar to the one described for causal ordering in Appendix B, but hopefully easier to understand.

Definition 14 (*Equilibrate* operator)**.** *Let* $M = \langle \boldsymbol{V}^t, \boldsymbol{E} \rangle$ *be a DBCM, and let* $V^t \in \boldsymbol{V}^t$*. Then Equilibrate*(M, V^t) *transforms* M *into another DBCM by applying the following rules:*

1. V^t *becomes an effect (and a constant), i.e., an equation in which* V^t *appears will be transformed into a prime equation where* V^t *is the prime variable.*
2. *All integral equations for* $\Delta^k V^t$*,* $k > 0$*, in* $\boldsymbol{E}$ *are removed.*
3. *The remaining occurrences of* $\Delta^k V^t$*,* $k > 0$*, in* $\boldsymbol{E}$ *are set to zero.*
4. *For each resulting equation* $0 := f(\boldsymbol{W}) + \epsilon$*, one* $W \in \boldsymbol{W}$ *that is not yet caused by another equation will become the effect such that* $W := f(\boldsymbol{W} \setminus W)$*.*

In general, the *Equilibrate* operator does not result in a unique equilibrium model, but in this dissertation I assume it does and, furthermore, is acyclic. This operator sounds more complex than it really is, so let me show a few examples to clarify. I will use the previously introduced example of the harmonic oscillator. Suppose x equilibrates, then all occurrences of v and a will be replaced with zero and x becomes an effect:

$$
\begin{aligned}
E_1: &\quad 0 := f_a(F_x, F_v, m) + \epsilon_a \\
E_2: &\quad x_c := f_x(F_x) + \epsilon_{F_x} \\
E_3: &\quad F_v := f_{F_v}(0) + \epsilon_{F_v} \\
E_4: &\quad m := \epsilon_m
\end{aligned}
$$

This, however, is not a valid DBCM yet because there is a zero at the left hand side of equation E_1. The only variable in equation E_1 that does not have causes yet is F_x, resulting in the following system:

$$
\begin{aligned}
E_1: &\quad F_x := f_a(F_v, m) + \epsilon_a \\
E_2: &\quad x_c := f_x(F_x) + \epsilon_{F_x} \\
E_3: &\quad F_v := f_{F_v}(0) + \epsilon_{F_v} \\
E_4: &\quad m := \epsilon_m
\end{aligned}
$$

As a more complex example, I will use the bathtub system used by Iwasaki [1988], which is also presented in Appendix B. A short introduction follows, but for more information please see the just mentioned resources. A bathtub is filling with rate F_{in} and has an outflow rate F_{out}. The change in depth D of the water in the tub is the difference between F_{in} and F_{out}. The change in pressure P on the bottom of the tub depends of the current depth and current pressure. The change in outflow rate is a function of valve opening V, pressure P and the current outflow rate F_{out}. The change in inflow rate and valve opening are determined exogenously. This results in the following DBCM:

$$
\begin{aligned}
E_1: &\quad \dot{F}_{\text{in}} := \epsilon_1 \\
E_2: &\quad \dot{D} := f_2(F_{\text{in}}, F_{\text{out}}) + \epsilon_2 \\
E_3: &\quad \dot{P} := f_3(D, P) + \epsilon_3 \\
E_4: &\quad \dot{V} := \epsilon_4 \\
E_5: &\quad \dot{F}_{\text{out}} := f_5(V, P, F_{\text{out}}) + \epsilon_5
\end{aligned}
$$

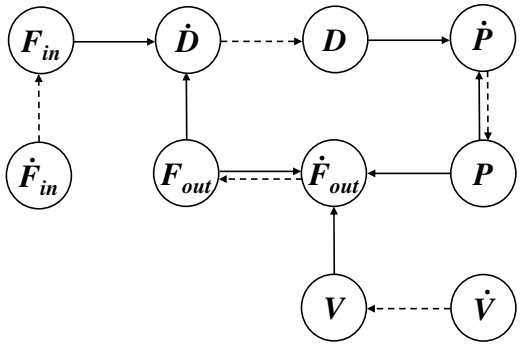

Figure 10: The dynamic graph of the bathtub system.

The dynamic shorthand graph of this system is displayed in Figure 10. Now, say that variable P equilibrates and we want to derive the causal structure of the resulting model. In this case, the left hand side of equation E_3 becomes 0, so we make P the effect since D is already determined by its integral equation (and the integral equation for P has been removed from the system by applying the *Equilibrate* operator). The resulting causal model

is described by the following equations:

$$
\begin{aligned}
E_1: &\quad \dot{F}_{\text{in}} := \epsilon_1 \\
E_2: &\quad \dot{D} := f_2(F_{\text{in}}, F_{\text{out}}) + \epsilon_2 \\
E_3: &\quad P := f_3(D) + \epsilon_3 \\
E_4: &\quad \dot{V} := \epsilon_4 \\
E_5: &\quad \dot{F}_{\text{out}} := f_5(V, P, F_{\text{out}}) + \epsilon_5
\end{aligned}
$$

The causal graph is displayed in Figure 11.

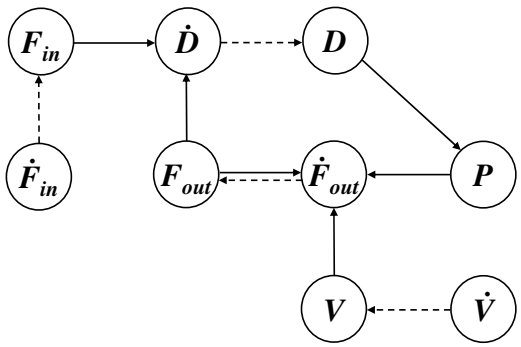

Figure 11: The dynamic graph of the bathtub system after P equilibrates.

Now suppose that all dynamic variables will be equilibrated. First, we set all the derivatives to zero:

$$
\begin{aligned}
E_1: &\quad 0 := \epsilon_1 \\
E_2: &\quad 0 := f_2(F_{\text{in}}, F_{\text{out}}) + \epsilon_2 \\
E_3: &\quad 0 := f_3(D, P) + \epsilon_3 \\
E_4: &\quad 0 := \epsilon_4 \\
E_5: &\quad 0 := f_5(V, P, F_{\text{out}}) + \epsilon_5
\end{aligned}
$$

Variables F_{in} and V are exogenous, so we put them at the left hand side of equation E_1 and E_4, respectively. Variable F_{in} will be the cause of variable F_{out} in equation E_2, and F_{in} and V will be the cause of P in E_5. This only leaves E_3 remaining, and since there is already a cause for P, P will have to cause D. This is the resulting set of equations:

$$
\begin{aligned}
E_1: &\quad F_{\text{in}} := \epsilon_1 \\
E_2: &\quad F_{\text{out}} := f_2(F_{\text{in}}) + \epsilon_2 \\
E_3: &\quad D := f_3(P) + \epsilon_3 \\
E_4: &\quad V := \epsilon_4 \\
E_5: &\quad P := f_5(V, F_{\text{out}}) + \epsilon_5
\end{aligned}
$$

The causal graph is shown in Figure 12.

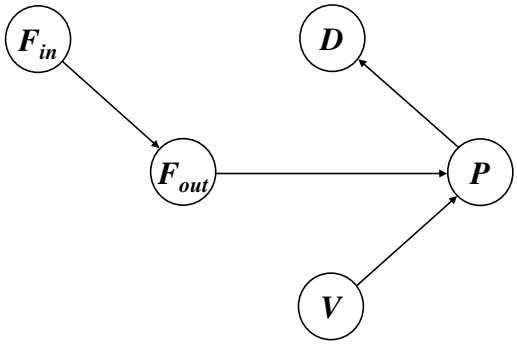

Figure 12: The causal graph of the bathtub example after equilibrating all the variables.

2.2.2 Manipulations

There are at least two different formalisms for performing manipulations, namely the *Do* operator and something what I will call restructuring. In this dissertation I will use the *Do* operator, but for completeness I will also shortly discuss restructuring.

2.2.2.1 The *Do* Operator The first type of manipulation that I will discuss is a standard operation in the causal discovery literature, namely the *Do* operator. This operation transforms one DBCM into another by replacing one equation in the model by another by setting the value of the variable that is manipulated to a constant. This also implies that all the derivatives of this variable will become zero.

Definition 15 (*Do* operator)**.** *Let* $M = \langle \boldsymbol{V}^t, \boldsymbol{E} \rangle$ *be a DBCM, and let* $V^t \in \boldsymbol{V}^t$*. Then* $\mathrm{Do}(M, V^t)$ *transforms* M *into another DBCM by applying the following rules:*

1. V^t *will be fixed to* $\hat{V}^t$*.*
2. *The equation for* $\Delta^p V^t$*,* $\Delta^p V^t$ *being the prime variable of* V^t*, will be removed from the system.*
3. *All integral equations for* $\Delta^k V^t$*,* $k > 0$*, in* $\boldsymbol{E}$ *are removed.*
4. *The remaining occurrences of* $\Delta^k V^t$*,* $k > 0$*, in* $\boldsymbol{E}$ *are set to zero.*

I will again use the simple harmonic oscillator example from the previous sections to show how the *Do* operator works. Suppose, that a manipulation is performed on variable x. This means that the mechanism that is responsible for the acceleration will no longer work, and neither do the integral equations. Instead, the value of x is determined directly:

$$
\begin{aligned}
E_1 : &\quad x := \hat{x} \\
E_2 : &\quad F_x := f_{F_x}(x) + \epsilon_{F_x} \\
E_3 : &\quad F_v := f_{F_v}(0) + \epsilon_{F_v} \\
E_4 : &\quad m := \epsilon_m
\end{aligned}
$$

Note that the *Do* operator replaces an equation completely, whereas the *Equilibrate* operator keeps the mechanism intact. In both cases all the derivatives become zero.

As another example, consider the bathtub model where all the variables have been equilibrated. Manipulating variable D will simply replace equation E_3 with $D := d$. The resulting causal graph is displayed in Figure 13. It amounts to cutting the arc $P \to D$.

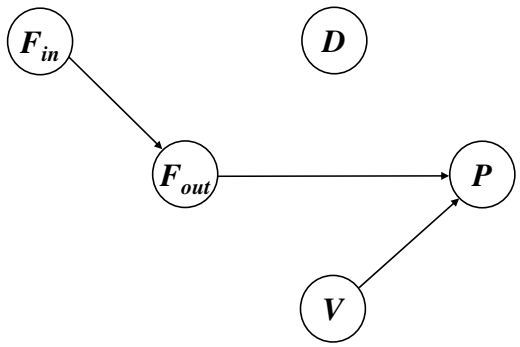

Figure 13: The causal graph after equilibrating all variables and then manipulating D.

Performing manipulations on an equilibrium graph can be problematic. For one, it becomes impossible to predict if the system becomes unstable. To make such predictions possible, the dynamic graph is required.

2.2.2.2 Restructuring A standard operation in the Iwasaki-Simon framework is transforming one valid model into another by making one endogenous variable exogenous and vice versa. The idea is that each equation forms a mechanism, and by changing the set of exogenous variables some of these mechanisms may reverse. I will discuss restructuring only in the context of SEMs, but it works for DBCMs as well.

Definition 16 (restructuring)**.** *A restructuring of a structural equation model is a transformation from one structural equation model to another by making one exogenous variable endogenous and vice versa, while keeping all the mechanisms intact.*

In a sense, restructuring is more than just a manipulation because a variable has to be

"released" as well, i.e., made endogenous. For example, consider the earlier introduced SEM:

$$
\begin{aligned}
E_1: &\quad A := \epsilon_A \\
E_2: &\quad B := \epsilon_B \\
E_3: &\quad C := f_C(A, B) + \epsilon_C \\
E_4: &\quad D := f_D(C) + \epsilon_D
\end{aligned}
$$

An example of restructuring would be to make D an exogenous variable instead of B. In this case there are two mechanisms, one for A, B, and C, and another for C and D. Therefore, in the new model D will cause C because of making D exogenous, and A and C will cause B, because B became endogenous. The resulting set of equations is displayed below, and the causal graph is displayed in Figure 14, which can be compared to the original graph in Figure 2.

$$
\begin{aligned}
E_1: &\quad A := \epsilon_A \\
E_2: &\quad B := f_B(A, C) + \epsilon_B \\
E_3: &\quad C := f_C(D) + \epsilon_C \\
E_4: &\quad D := \epsilon_D
\end{aligned}
$$

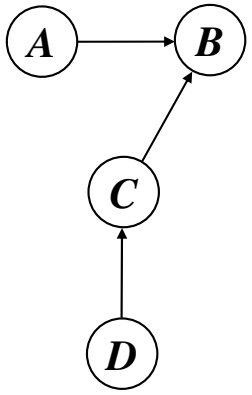

Figure 14: The causal graph after restructuring.

2.2.3 Equilibration-Manipulation Commutability

One of the fundamental purposes of causal models is using them to predict the effects of manipulating various components of a system. It has been argued by Dash [2003, 2005] that the *Do* operator will fail when applied to an equilibrium model, unless the underlying dynamic system obeys what he calls *Equilibration-Manipulation Commutability (EMC)*, a principle which is illustrated by the graph in Figure 15. In this figure, a dynamic system S, represented by a set of differential equations, is depicted at the top. S has one or more equilibrium points such that, under the initial exogenous conditions, the equilibrium model $\tilde{S}$, represented by a set of equilibrium equations, will be obtained after sufficient time has passed. There are thus two approaches for making predictions of manipulations on S on time-scales sufficiently long for the equilibrations to occur. One could start with $\tilde{S}$ and apply the *Do* operator to predict manipulations. This is path A in Figure 15, and is the approach taken whenever a causal model is built from data drawn from a system in equilibrium. Alternatively, in path B the manipulations are performed on the original dynamic system which is then allowed to equilibrate; this is the path that the actual system takes. The EMC property is satisfied if and only if path A and path B lead to the same causal structure.

Dash [2003] proved that there are conditions under which the causal predictions $\tilde{\hat{S}}$ and $\hat{\tilde{S}}$ are the same, and conditions under which they are different. First, I introduce the concept of a feedback set. A feedback for a variable contains all variables that are both ancestors and descendents.

Definition 17 (feedback set)**.** *The feedback set* $\boldsymbol{Fb}$ *of a variable* V *is given by*

$$\boldsymbol{Fb}(V) = \boldsymbol{Anc}(V) \cap \boldsymbol{Desc}(V),$$

where $\boldsymbol{Anc}(V)$ *denotes the set of ancestors in a DBCM graph, and* $\boldsymbol{Desc}(V)$ *denotes the set of descendents in a DBCM graph.*

I will now state a theorem that provides a sufficient condition for EMC violation.

Theorem 1 (EMC violation)**.** *Let* $M = \langle \boldsymbol{V^t}, \boldsymbol{E} \rangle$ *be a DBCM and let* $M_{\tilde{V}^t} = \langle \boldsymbol{V^{t'}}, \boldsymbol{E'} \rangle$ *be the same model in which* $V^t \in \boldsymbol{V^t}$ *is equilibrated. If there exists any* $Y^t \in \boldsymbol{Fb}(V^t)_M$ *such that* $Y^t \in \boldsymbol{V^{t'}}$*, then* $Do(M_{\tilde{V}^t}, Y^t) \neq Equilibrate(Do(M, Y^t), V^t)$.

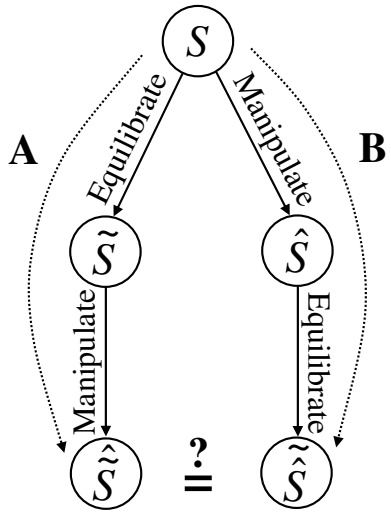

Figure 15: Equilibration-Manipulation Commutability provides a sufficient condition for an equilibrium causal graph to correctly predict the effect of manipulations.

In words, this means that if any of the variables in the feedback set before equilibration are still in the model after equilibration, EMC will be violated. A sufficient condition for EMC obeyence is given in the next theorem.

Theorem 2 (EMC obeyence)**.** *Let M and $M_{\tilde{V}^t}$ be defined as in the previous theorem, and let $\Delta^n V_i^t \in \boldsymbol{V^t}$ be the prime variable of $V_i^t \in \boldsymbol{V^t}$. If $V_i^t \in \boldsymbol{Pa}(\Delta^n V_i^t)_M$, then $Do(M_{\tilde{V}^t}, Y^t) = Equilibrate(Do(M, Y^t), V^t)$.*

This theorem says that if a prime variable has as cause any of its lower order derivatives, the EMC condition will be obeyed. Proofs of these two theorems can be found in Dash [2003].

As an example of a system that obeys the EMC condition, again consider a body of mass m dangling from a damped spring. The mass will stretch the spring to some equilibrium position $x = mg/k$ where k is the spring constant. As we vary m and allow the system to come to equilibrium, the value of x gets affected according to this relation. The equilibrium

causal model $\tilde{S}$ of this system is simply $m \rightarrow x$. If one were to manipulate the spring directly and stretch it to some displacement $x = \hat{x}$, then the mass would be independent of the displacement, and the correct causal model is obtained by applying the Do operator to this equilibrium model.

Alternatively, one could have started with the original system S of differential equations of the damped simple-harmonic oscillator by explicitely modeling the acceleration $a = mg - kx - \alpha v$, where α is the dampening constant, and the velocity v. S can likewise be used to model the manipulation of x by applying the Do operator to a, v, and x simultaneously, ultimately giving the same structure as was obtained by starting with the equilibrium model.

To exemplify EMC violation, I will return to the example of the bathtub introduced in the previous section. Suppose that variables P and F_{out} are already equilibrated and the next variable to equilibrate is D. The causal graph of this situation is displayed in Figure 16. First, we establish the variables in the feedback set of D:

$$\boldsymbol{Fb}(D) = \{P, F_{\text{out}}, \dot{D}\} \ .$$

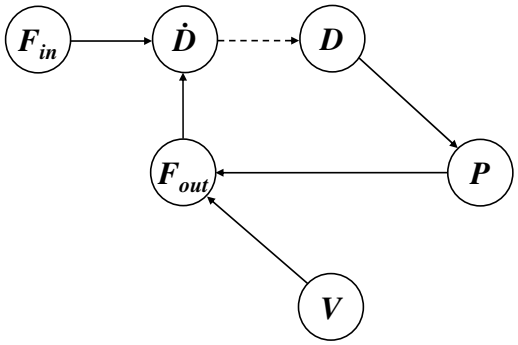

Figure 16: The causal graph of the bathtub example before equilibrating D.

Figure 17 shows the resulting graph after equilibration. It is easy to see that P and F_{out}, which are in the feedback set, also appear in that graph. Therefore, the EMC condition is violated and it is not safe to use the graph of Figure 17 to make manipulation predictions.

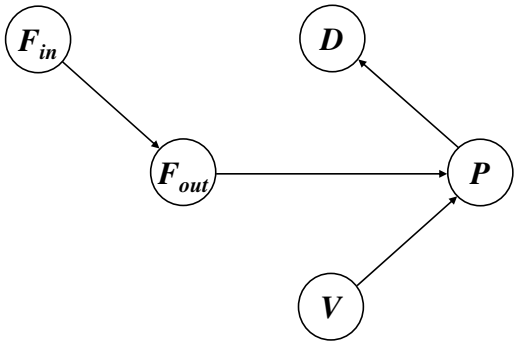

Figure 17: The causal graph of the bathtub example after equilibrating D.

If we would manipulate variable D, the arc from P to D in Figure 17 is cut by the *Do* operator. The correct way would be to manipulate D in the dynamic graph and then equilibrate. This results in the graph displayed in Figure 18, which is clearly different from the one in Figure 17.

The reason that the EMC condition is not violated when P and F_{out} equilibrate is that their feedback sets only consist of their corresponding derivative variable:

$$\boldsymbol{Fb}(P) = \{\dot{P}\},$$
$$\boldsymbol{Fb}(F_{\text{out}}) = \{\dot{F}_{\text{out}}\}.$$

These variables are called *self-regulating*, because the derivative is caused by its own variable. The resulting graphs after equilibration are shown in Figure 11 and 16. Manipulating these graphs using the *Do* operator will result in the same graph as manipulating the original dynamic graph first and then equilibrating.

One of the fundamental purposes of causal models is using them to predict the effects of manipulating various components of a system. The EMC violation theorem stated previously showed that the *Do* operator will fail when applied to an equilibrium model. Unfortunately,

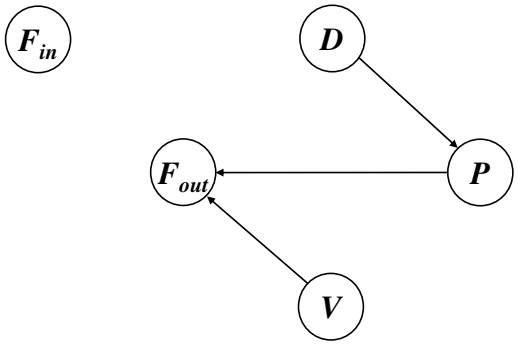

Figure 18: The causal graph of the bathtub example after first performing a manipulation on D and then equilibrating.

this fact renders most existing causal discovery algorithms unreliable for reasoning about manipulations, unless the details of the underlying dynamics of the system are explicitly represented in the model. Most classical causal discovery algorithms in AI make use of the class of independence constraints found in the data to infer causality between variables, assuming the faithfulness assumption [e.g., Spirtes et al., 2000, Pearl and Verma, 1991, Cooper and Herskovits, 1992]. These methods will not be guaranteed to obey EMC if the observation time-scale of the data is long enough for some process in the underlying dynamic system to go through equilibrium.

In fact, violation of the EMC condition can be seen as a particular case of violation of the *faithfulness* assumption. Faithfulness is the converse of the Markov condition, and it is the critical assumption that allows structure to be uncovered from independence relations. It has been argued [e.g., Spirtes et al., 2000] that the probability of a system violating faithfulness due to chance alone has Lebesgue measure 0. However, when a dynamic system goes through equilibrium, *by definition*, faithfulness is violated. For example, if the motion of the block in the earlier mentioned example reaches equilibrium, then by definition, the equation $a = (F_x + F_v + mg)/m$ becomes $0 = F_x + F_v + mg$. This means that the values of

the forces acting on the block are no longer correlated with the value of a, even though they are direct causes of a.

Definition 18 (faithfulness)**.** *A probability distribution $P(\boldsymbol{V})$ obeys the faithfulness condition with respect to a directed acyclic graph G over $\boldsymbol{V}$ if and only if for every conditional independence relation entailed by P there exists a corresponding d-separation condition entailed by G:* $(X \perp\!\!\!\perp Y \mid \boldsymbol{Z})_P \Rightarrow (X \perp\!\!\!\perp_d Y \mid \boldsymbol{Z})_G$.

The EMC condition has two implications for DBCM learning. First, if we are interested in the original DBCM, we must learn models from time-series data with temporal resolution small enough to rule out any equilibration occuring. Second, and more astonishing, is that even if we are only concerned about the long-time-scale or equilibrium behavior of a system, if we desire a model that will allow us to correctly predict the effects of manipulation, we still must learn the fine time-scale model, unless we get lucky and are dealing with a system that just happens to have the same structure when it passes through an equilibrium point. Dash [2003] discusses some methods to detect such systems and refers to them as obeying the EMC condition.

2.3 LEARNING

Even if one is only interested in the long-term equilibrium behavior of a system, it is still necessary to learn the system's underlying dynamics in order to do causal reasoning. As was explained in the previous section, Dash [2003, 2005] has demonstrated convincingly that the *Do* operator will fail when applied to an equilibrium model unless the underlying dynamic system happens to obey what he calls *Equilibration-Manipulation Commutability (EMC)*. Therefore, one must in general start with a non-equilibrated dynamic model in order to reason about manipulations on the equilibrium model correctly. Motivated by that caveat, in this section I present a novel approach to causal discovery of dynamic models from time series data. The approach uses the representation of dynamic causal models developed in the previous sections. I present an algorithm that exploits this representation within a constraint-based learning framework by numerically calculating derivatives and learning instantaneous relationships. I argue that due to numerical errors in higher order derivatives, care must be taken when learning causal structure, but I show that the DBCM representation reduces the search space considerably, allowing us to forego calculating many high-order derivatives. In order for an algorithm to discover the dynamic model, it is necessary that the time-scale of the data is much finer than any temporal process of the system, as was argued in the previous section. In the next chapter, I show that my approach can correctly recover the structure of a fairly complex dynamic system, and can predict the effect of manipulations accurately when a manipulation does not cause an instability. To the best of my knowledge, this is the first causal discovery algorithm that has demonstrated that it can correctly predict the effects of manipulations for a system that does not obey the EMC condition.

There have been previous approaches for learning dynamic causal models. Among them are Dynamic Bayesian Networks (DBNs) [Friedman et al., 1998a], Granger causality [Granger, 1969, Engle and Granger, 1987, Sims, 1980], and vector autoregression models [Swanson and Granger, 1997, Demiralp and Hoover, 2003]. DBCM learning will be compared to the latter two approaches in the last section of this chapter. Effectively, while these methods are general and consider arbitrary relations between variables across time, they do not exploit some underlying constraints if the underlying system is governed by differential

equations.

DBCMs assume that all causation works in the same way as causality in mechanical systems, i.e., all causation *across time* is due to integration. This restriction represents a tradeoff between expressibility and tractability. On the one hand, DBCMs are able to represent all mechanical systems and a large class of non-first-order Markovian graphs that can also be converted to DBCMs. On the other hand, more restricted structure of the DBCMs' guarantees that a learned model will be first-order Markovian. Also, DBCMs are in principle easier to learn because, even if some required derivatives are unobserved in the data, at least we know something about these latent variables that are required to make the system Markovian.

The algorithm, which I will call DBCM Learner, does not assume that all relevant derivatives of the system are known, and conducts an efficient search to find them, treating them as latent variables. However, we exploit the fact that these derivatives have fixed relationships to some known variables and so are easier to find than general latent variables. The derivatives are calculated in the following way:

$$\dot{x}^t = x^{t+1} - x^t$$
$$\ddot{x}^t = \dot{x}^{t+1} - \dot{x}^t$$

Higher order derivatives are obtained in a similar way. The DBCM Learner is also robust in the sense that it avoids calculating higher-order derivatives unless they are required by the model, thus avoiding mistakes due to numerical errors. I prove that the algorithm is correct up to the correctness of the underlying conditional independence tests.

The DBCM Learner can be thought of as two separate steps: (1) detecting prime (and integral) variables, and (2) learning the contemporaneous structure. Theorems and examples will be given in the next sections, but the proofs will be deferred to Appendix C.

2.3.1 Detecting Prime Variables

Detecting prime variables is based on the fact that by definition there are no edges between prime variables in two consecutive time slices. Conversely, integral variables always have an edge from themselves in the previous time slice. This is clearly illustrated by looking at the unrolled version of a DBCM graph, such as the one for the harmonic oscillator displayed in Figure 23. There are direct edges between x^0 and x^1, and v^0 and v^1, but not a^0 and a^1. This follows from the way integral equations are defined. The following theorem exploits this fact to find prime variables.

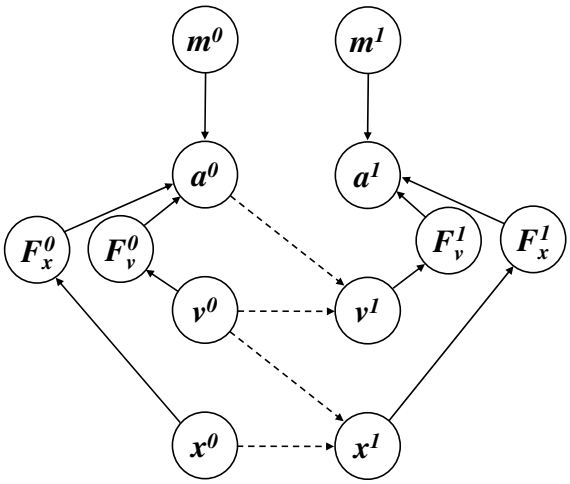

Figure 19: The unrolled version of the simple harmonic oscillator where it is clearly visible that integral variables are connected to themselves in the previous time slice, and prime variables are not.

Theorem 3 (detecting prime variables)**.** *Let* $\boldsymbol{V^t}$ *be a set of variables in a time series faithfully generated by a DBCM and let* $\boldsymbol{V^t_{all}} = \boldsymbol{V^t} \cup \Delta \boldsymbol{V^t}$*, where* $\Delta \boldsymbol{V^t}$ *is the set of all differences of* $\boldsymbol{V^t}$*. Then* $\Delta^j V_i^t \in \boldsymbol{V^t_{all}}$ *is a prime variable if and only if*

1. *There exists a set* $\boldsymbol{W} \subset \boldsymbol{V^t_{all}} \setminus V_i^t$ *such that* $(\Delta^j V_i^{t-1} \perp\!\!\!\perp \Delta^j V_i^t \mid \boldsymbol{W})$.
2. *There exists no set* $\boldsymbol{W'} \subset \boldsymbol{V^t_{all}} \setminus V_i^t$ *such that* $(\Delta^k V_i^{t-1} \perp\!\!\!\perp \Delta^k V_i^t \mid \boldsymbol{W'})$ *for* $k < j$.

The theorem basically states that by conditioning on a subset of variables in time slice t, we can never break the edges between integral variables, but we can always make V_i^{t-1} independent of V_i^t if it is a prime variable for the following reasons:

1. V_i^{t-1} and V_i^t can only be dependent if there is an influence that goes through one of the integral variables in time slice t.
2. By conditioning on all integral variables in time slice t this influence can be blocked. Also, integral variables can only have outgoing edges to variables in the same time slice, so no v-structures will be "enabled".

To illustrate this process, again look at the simple harmonic oscillator in Figure 23. Obviously, the direct connection between v^0 and v^1, and x^0 and x^1 cannot be broken because there is a direct edge. a^0 and a^1, however, can be made independent by conditioning on, for example, v^1 and x^1.

After the prime variables have been detected, the integral variables are implicit and can be retrieved because the set of integral variables for any variable V_i are given by $\Delta^k V_i$, $0 \leq k < j$, when $\Delta^j V_i$ is the prime variable of V_i. If the prime variable is of zeroth order it is a static variable, but it is detected in exactly the same way.

One might argue that because there are deterministic relationships (the integral equations) in a DBCM, it is impossible to apply faithfulness to such a model. However, note that all the variables in the conditioning set are from $\boldsymbol{V^t}$ and, therefore, there are no deterministic relationships in the set of variables $\{\Delta^k V_i^{t-1}\} \cup \boldsymbol{V^t}$. $\Delta^k V_i^{t-1}$ and $\Delta^{k+1} V_i^{t-1}$ deterministically cause $\Delta^k V_i^t$, but $\Delta^{k+1} V_i^0$ is never in the conditioning set and can be thought of as a noise term.

2.3.2 Learning Contemporaneous Structure

Once we have found the set of prime variables, learning the contemporaneous structure becomes a problem of learning a time-series model from *causally sufficient* data (i.e., there do not exist any latent common causes). In addition to discovering the latent variables in the data, we also know that there can be no contemporaneous edges between two integral variables, and integral variables can have only outgoing edges. We can thus restrict the search space of causal structures. The next theorem shows we will learn the correct structure.

Theorem 4 (learning contemporaneous structure)**.** *Let* $\boldsymbol{V^t}$ *be a set of variables in a time series faithfully generated by a DBCM and let* $\boldsymbol{V^t_{all}} = \boldsymbol{V^t} \cup \Delta \boldsymbol{V^t}$*, where* $\Delta \boldsymbol{V^t}$ *is the set of all differences of* $\boldsymbol{V^t}$ *that are in the DBCM. Then there is an edge* $V^t_i - V^t_j$ *if and only if there is no set* $\boldsymbol{W} \in \boldsymbol{V^t_{all}} \setminus V^t_i, V^t_j$ *such that* $(V^t_i \perp\!\!\!\perp V^t_j \mid \boldsymbol{W})$*.*

Theorem 4 shows that we can learn the contemporaneous structure from time-series data despite the fact that data from time-to-time is not independent. This is because, by construction, we know the set of integral variables in time t will render $\boldsymbol{V^t} \setminus \boldsymbol{V^t_{int}}$ independent of $\boldsymbol{V^{t-1}}$, where $\boldsymbol{V^t_{int}}$ is the set of integral variables. Furthermore, $\boldsymbol{V^t_{int}}$ is precisely the set of variables we do not need to search for structure between, because it is specified by the definition of DBCMs.

For example, consider the simple harmonic oscillator in Figure 23 again. Variables F^1_v and F^1_x are correlated, because v^0 is a common cause, but by conditioning on x^1 or v^1, or both, F^1_v and F^1_x become independent.

As before, faithfulness is not an issue here because integral equations are across time and here we are only considering within time-slice causality. But it is assumed that the contemporaneous structure does not change over time.

2.3.3 The DBCM Learner

The previous sections provided theorems that showed it is possible to learn DBCMs from data. In this section, these theorems are translated into a concrete algorithm. Although the theorems made a distinction between finding prime variables and finding the contemporane-

ous structure, the algorithm does both at the same time for efficiency reasons. However, for better results, separating the search for prime variables from the search for the contemporaneous structure should be preferred. First, I will explain how the DBCM Learner works and afterwards I present an example for illustration. The DBCM Learner uses the PC algorithm internally, and that algorithm is covered in Appendix A.

Algorithm 1 (DBCM Learner)**.**

Input: *A time series T with variables $\boldsymbol{V}$ and a maximum derivative $k_{\max}$.*

Output: *A DBCM pattern.*

1. *Initialize $k = 0$ and U as an empty undirected graph.*
2. *Add $\Delta^k V_i$ to the undirected graph U if no prime variable has been found for $V_i \in \boldsymbol{V}$ yet.*
3. *Connect edges from the newly added variables to all the variables already in the model. Also connect the newly added variables to themselves.*
4. *Run the standard PC algorithm on undirected graph U by using data from the time series. Each time slice is considered to be a record.*
5. *For each variable check if there is a prime variable. This is done by checking if $\Delta^m V_i^{t-1}$ is independent of $\Delta^m V_i^t$, for all $0 \leq m \leq k$, by conditioning on all direct neighbors of $\Delta^m V_i^t$ in U. Every two consecutive time slices, i.e., $t-1$ and t, are combined into one record. If a prime is found, it will be added to $\boldsymbol{V}_{\mathbf{pr}}$. All derivatives higher than the prime variable are removed from the model.*
6. *$k = k + 1$.*
7. *Goto step 2 if $k \leq k_{\max}$.*
8. *Remove all edges between integral variables.*
9. *Add the dashed integral edges.*
10. *Orient all edges from integral variables as outgoing.*
11. *Orient all other edges according to the rules in PC.*
12. *Return the resulting DBCM pattern.*

The DBCM Learner looks for prime variables by conditioning on all current derivatives in the model, and the derivatives are added stepwise. I will illustrate this process by using the simple harmonic oscillator. Figure 20 shows how the prime variables are detected. On the

far left is the initial undirected fully connected graph that is the starting point. At the center left, all edges have been removed except the one between x and F_x because the true model has an edge, and the one between x and F_v because v is not included yet so they cannot be made conditionally independent. There are no edges between the other variables, because they form a v-structure on a which is not included in the model yet and by conditioning on x all the common causes are blocked. All variables are determined to be primes, except for x and F_v, because x is really a prime variable and F_v^{t-1} is not independent of F_v^t since v has not been added to the model yet. This is corrected in the next step, center right, and there is also an edge introduced between v and F_v. Variables x and v are directly dependent because of a common cause in the previous time slice. The last step adds a to the model and the correct edges have been identified, depicted far right.

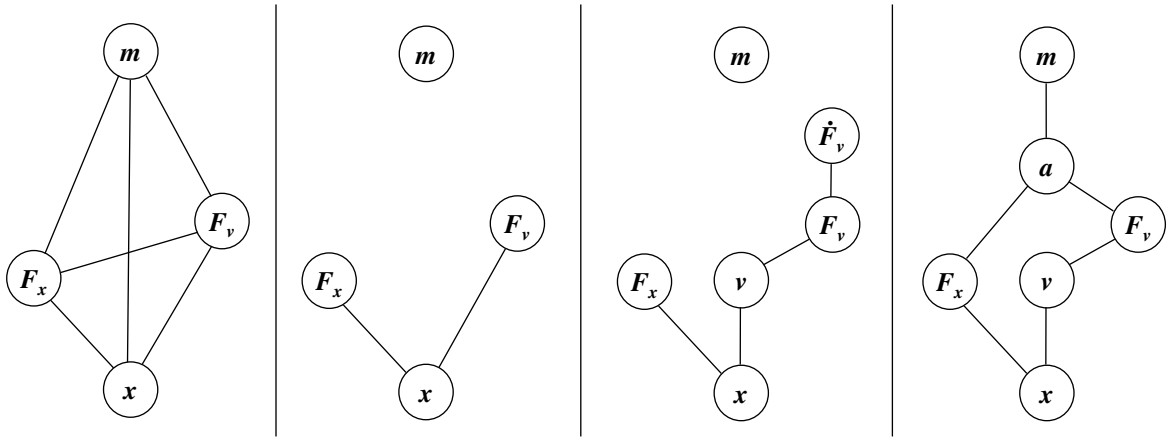

Figure 20: Far left: The starting graph. Center left: After the first iteration. Center right: After the second iteration. Far right: The final undirected graph.

Figure 21 shows the final steps. At the left the integral edges are added from a to v and v to x. In the center all edges from integral variables are oriented as outgoing. At the right the remaining edges are oriented using the rules for edge orientation in PC.

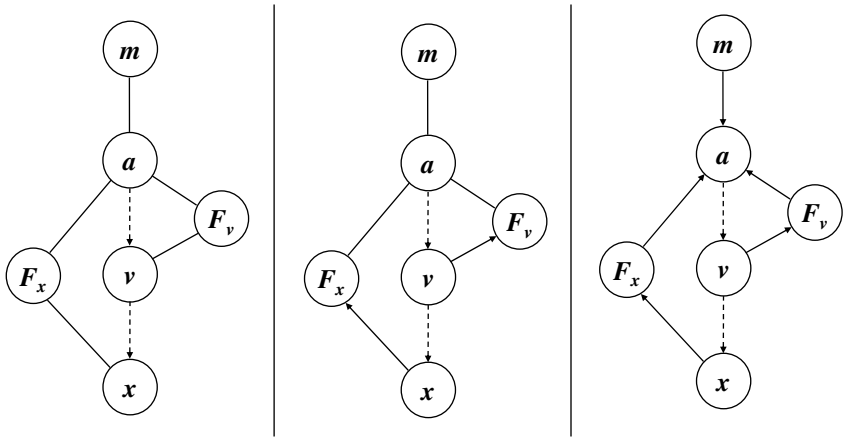

Figure 21: Left: Orientation of the integral edges. Center: Orient edges from integral variables as outgoing. Right: Orient the remaining edges.

The previous section showed the implications of the EMC condition. First, if a variable is *self-regulating*, meaning that $X \in Pa(\Delta^m X)$, where $\Delta^m X$ is the prime variable, then when X is equilibrated, the parent set of X and the children set of X are unchanged. Thus with respect to manipulations on X, the EMC condition is obeyed. Second, a sufficient condition for the violation of EMC comes when the set of feedback variables of some X is nonempty in the equilibrium graph. In this case, there will always exist a manipulation that violates EMC.

Since the DBCM learner is not guaranteed to find the orientation of every edge in the DBCM structure, it is a valid question to ask if the method is guaranteed to be useful for detecting EMC violation. The following two theorems show that it is. Theorem 5 shows that we can always identify whether or not a variable is self-regulating, and Theorem 6 shows that we can always identify the feedback set of every variable. Both theorems rely on the presence of an accurate independence oracle.

Theorem 5. *Let D be a DBCM with a variable X that has a prime variable $\Delta^m X$. The pdag returned by Algorithm 1 with a perfect independence oracle will have an edge between X and $\Delta^m X$ if and only if X is self-regulating.*

Theorem 6. *Let G be the contemporaneous graph of a DBCM. Then for a variable X in G, $\boldsymbol{Fb}(X) = \emptyset$ if and only if for each undirected path P between X and $\Delta^m X$, there exists a v-structure $P_i \to P_j \leftarrow P_k$ in G such that $\{P_i, P_j, P_k\} \subset \boldsymbol{P}$.*

The proofs are given in Appendix C. Because a correct structure discovery algorithm will recover all v-structures, Theorem 6 tells us how to identify all feedback variables. In fact, this theorem does not make use of the fact that the path terminates on a prime variable, so we can in fact determine whether or not a directed path exists from an integral variable to any other variable in the DBCM.

2.4 ASSUMPTIONS

Summarizing, here is a list of assumptions that have to be satisfied for the DBCM Learner to work properly:

- The standard assumptions in causal discovery that are given in Appendix A.
- The underlying model is a DBCM. This implies that all causation across time is due to integral equations and there is no contemporaneous causation.
- No equilibrations should have occurred in the data, otherwise we would be learning an equilibrium model.
- Non-constant error terms over time, i.e., the error terms are resampled in each time step.
- No latent confounders.

I will now briefly comment on two of these assumptions, namely non-constant error terms and no latent confounders.

2.4.1 Non-Constant Error Terms

Instead of assuming that all error terms across time are independent, sometimes the assumption is made that the error terms are constant over time, i.e., they are sampled only once and then kept fixed. If that is the case, the DBCM Learner will break down, because it requires noise to properly learn structure. One possible fix would be to obtain multiple time series in which the error terms are resampled, and then select one sample from each of those time series to combine them into a time series where all the error terms are independent of each other.

2.4.2 No Latent Confounders

DBCM learning assumes that all common causes of at least two variables are included in the data set. PC has a counterpart that is able to learn causal structure for hidden variables, namely the FCI algorithm [Spirtes et al., 2000]. Applying this to DBCMs is not straightforward, because a missing a prime variable could lead to unexpected results. For

example, consider the network shown at the left hand side of Figure 22. Now suppose x is a hidden variable and only data for y is available. The resulting learned model is displayed at the right of Figure 22. Because we cannot condition on x anymore, it will look like as if y is a self-regulating variable because y will roughly follow the same time-path as x.

At this time, I do not know if there exist any guarantees that can be given when there are hidden variables. It could, for example, be that guarantees can only be given when only static variables are among the hidden variables, which sounds plausible because it does not seem to affect finding prime variables.

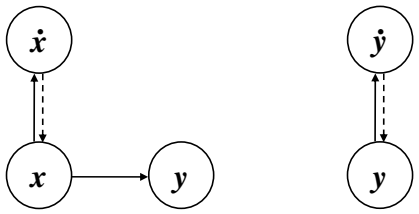

Figure 22: Left: Original model. Right: Learned model if x is a hidden variable.

2.5 COMPARISON TO OTHER APPROACHES

Before comparing DBCM learning to other approaches, I will first briefly introduce two prominent approaches that also learn models from time series data. They are Granger causality and vector autoregression.

2.5.1 Granger Causality

Granger causality [Granger, 1969, 1980] is a technique used in determining if one time series is useful in forecasting another time series. Clive Granger, winner of the Nobel Prize in Economics, argued that by making use of the implicit role of time, there is an interpretation of a set of tests that reveal something about causality.

Definition 19 (Granger causality)**.** *A time series X is said to Granger cause time series Y if and only if Y_{t+1} is not independent of $X_{1...t}$ (all lags on X) conditional on $Y_{1...t}$ (all lags on Y).*

Usually, F-tests are used to test for conditional independence. Granger causality is not considered to be true causality, because it only looks at two variables at a time. It also excludes the possibility of contemporaneous causality. The procedure is only applicable on pairs of variables. A similar procedure involving more variables can be applied with vector autoregression, which is discussed next.

2.5.2 Vector Autoregression

Several procedures have been developed that take as input a time series and then deduce a causal structure. One prominent approach to discovery of causality in time series are Vector AutoRegression (VAR) models [Sims, 1980] that embed the principle of Granger causality. An alternative, but related, approach uses dynamic factor models [Moneta and Spirtes, 2006]. I will now briefly discuss VAR models.

A (reduced) pth order VAR is defined as follows. Let k be the number of variables, y_t a $k \times 1$ vector which has as the ith element $y_{i,t}$ the time t observation of variable y_i. Then the

VAR is given by

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t \ ,$$

where c is a $k \times 1$ vector of constants, A_i is a $k \times k$ matrix and e_t is a $k \times 1$ vector of error terms. The following conditions hold on the error terms:

1. $\mathrm{E}(e_t) = 0$
2. $\mathrm{E}(e_t e_t') = \Omega$, where Ω is the contemporaneous covariance matrix of error terms.
3. $\mathrm{E}(e_t e_{t-k}') = 0$ for any non-zero k, i.e., there is no correlation across time.

Matrices A_i are usually estimated using the ordinary least squares method. The covariance matrix of resulting error terms can have non-zero off-diagonal elements, thus allowing non-zero correlation between error terms. Correlated error terms lead to difficulties when shocking (manipulating) the error variables, because it will simultaneously deliver correlated shocks to other variables as well. An important problem in econometrics is to transform the VAR into a Structural VAR (SVAR), in which the error terms are uncorrelated. This structural equation form cannot be uniquely identified from a VAR and it requires a causal ordering on the contemporaneous variables. This causal ordering used to be determined by background knowledge, but lately standard causal learning algorithms have been used to establish this ordering [Swanson and Granger, 1997, Demiralp and Hoover, 2003], and even approaches to non-linear systems have been proposed [Chu and Glymour, 2008].

2.5.3 Discussion

Given the prevalence of real-world systems driven by their underlying dynamics, it is important to have a learning algorithm that learns the DBCM representation directly. In these cases, we know that the latent derivative variables have a fixed structure which we can search for. Furthermore, once all those variables are found, we can restrict our structure search to contemporaneous structure with additional constraints on directionality of edges. This contrasts with other methods such as dynamic SEMs, Granger causality, VAR models, dynamic Bayesian networks [Friedman et al., 1998a] and the Granger-based causal graphs of [Eichler and Didelez, 2007], that allow arbitrary edges to exist across time. For many real physical

systems this representation is too general: allowing things like contemporaneous causal cycles, causality going backward in time and arbitrary cross-temporal causation. DBCMs, by contrast, assume that all causation works in the same way as in mechanical systems, i.e., all causation *across time* is due to integration. This restriction represents a tradeoff between expressibility and tractability. On the one hand, DBCMs are able to represent all mechanical systems and a large class of non-first-order Markovian graphs and guarantees that a learned model will be first-order Markovian. DBCMs are in principle easier to learn because, even if some required derivatives are unobserved in the data, at least we know something about these latent variables that are required to make the system Markovian.

When confronted with data that has not made all relevant derivatives explicit, the distinction between DBCMs and the other approaches becomes glaring. Whereas a DBCM discovery algorithm attempts to search for and identify the latent derivative variables, other approaches would try to marginalize them out. The idea of the marginalization is the following. In a data set, usually only the variables are included and no derivatives. The DBCM Learner tries to find these hidden variables. Granger causality and VAR do not search for these hidden variables, effectively learning a model where these variables have been marginalized out. One might have suspected that there is not much difference. For example, one might expect that a second order differential equation would simply result in a second-order Markov model when the derivatives are marginalized out. Unfortunately, that is not the case, because the causation among the derivatives forms an infinite chain into the past. Thus any approach that tries to marginalize out the derivatives must include infinite edges in the model. For example, consider the graph in Figure 23. If we marginalize out all the derivatives of x, i.e., v and a, then all parents of a in time-slice i of the DBCM are parents of x for all time slices $j > i + 1$. See the Figure for a more specific example. Thus, the benefits of using the DBCM representation are not merely computational, but in fact without learning the derivatives directly, the correct model does not even have a correct finite representation. In the next chapter empirical evidence is shown that confirms this.

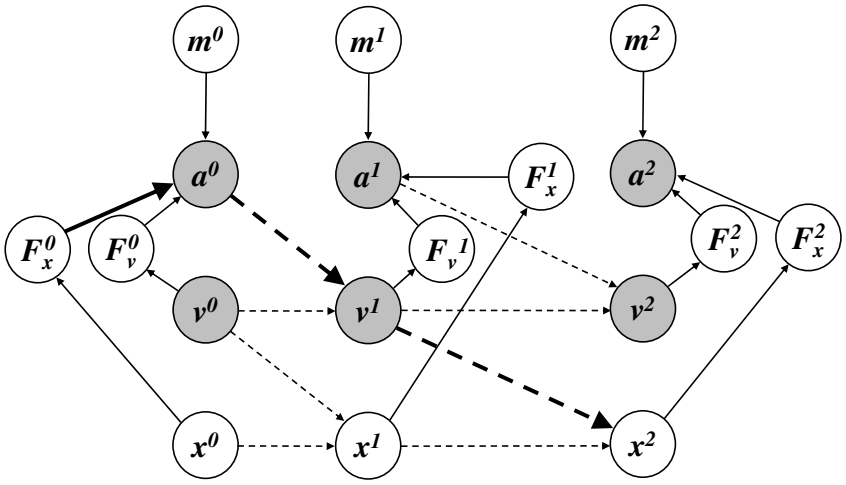

Figure 23: Marginalizing out the derivatives v and a results in higher-order Markovian edges to be present (e.g., $F_x^0 \rightarrow x^2$). Trying to learn structure over this marginalized set directly involves a larger search-space.

3.0 EXPERIMENTAL RESULTS

To test the DBCM Learner in practice, several experiments were performed. They can be divided into two different parts, namely the experiments where a gold standard network was relearned to assess the performance of the DBCM Learner, and the experiments where the truth is unknown or only partially known and DBCM learning is used for exploring.

3.1 HARMONIC OSCILLATORS

First, I generated data from real physical systems where ground truth is known and are representative of the type of systems found in nature. The initial idea was to generate random graphs, sample data from them, and try to relearn them. It turns out, however, that learning DBCMs from such data is problematic. The main issue is that the generated systems are frequently unstable, so the numbers get very large compared to the noise terms causing many violations of faithfulness to occur in the data. Instead, I focused on several different causal graphs with parameters that lead to a stable system.

It was difficult to establish a baseline approach because there are few methods available that deal with temporal systems, causality, latent and continuous variables, all at the same time. I resorted to using a modified form of the PC algorithm [Spirtes et al., 2000] and a Bayesian scoring algorithm for the comparisons. And as was discussed in Section 2.5, there does not exist a suitable baseline method that is even in principle able to correctly learn the simple harmonic oscillator model of Figure 8. If one tries to learn causal relations with the latent variables marginalized out, an infinite-order Markov model results (Figure 23). Thus, the validation of my method is complicated by the fact that there is no way to measure the

correctness of the models produced by existing methods. Even methods such as the FCI algorithm [Spirtes et al., 2000] which attempt to take into consideration latent variables, would still result in an infinite Markov model because it does not try to isolate and learn the latents and structure between them and the observables. Methods such as the structural EM algorithm [Friedman et al., 1998b] might be appropriate, but they would have to be adapted for temporal data.

To verify the practical applicability of the method, I tested it on models of two physical systems, namely a simple harmonic oscillator (Figure 8) and a more complex coupled harmonic oscillator (Figure 24). For both systems I selected the parameters in the models in such a way that they were stable, i.e., returned to equilibrium and produced measurements within reasonable bounds. I generated 100 data sets of 5,000 records for each system. All but the integral equations had a noise term associated with them, just as DBCMs assume.

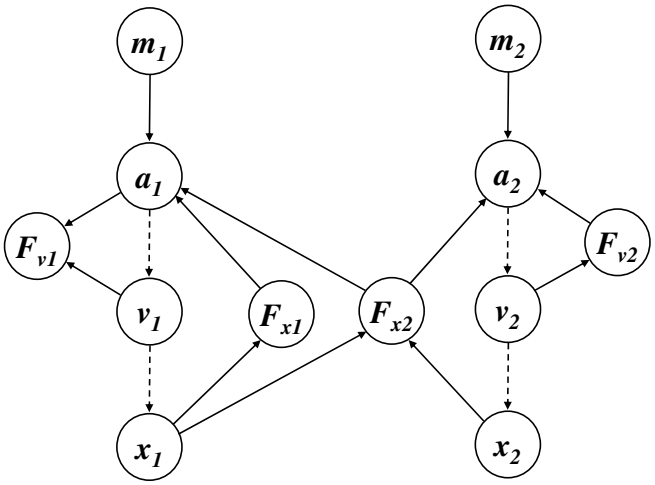

Figure 24: Causal graph of the coupled harmonic oscillator.

The DBCM Learner conducts a search for prime variables and the contemporaneous structure at the same time for efficiency. In this series of experiments, I will split the algorithm in two steps, namely finding the derivatives and finding the contemporaneous structure. The output of the first stage, a set of prime variables, will be used as the input

to the second stage. Looking for prime variables is done by directly applying Theorem 3 and finding the contemporaneous structure is done by directly applying Theorem 4. As a reminder, here is how the derivatives were calculated:

$$\dot{x}^t = x^{t+1} - x^t$$
$$\ddot{x}^t = \dot{x}^{t+1} - \dot{x}^t$$

Higher order derivatives were obtained in a similar way.

The baselines that I chose were to use PC and a greedy Bayesian approach on a data set with all differences up to some maximum $k_{max} = 3$ calculated a priori, and to interpret the structure as best I could as a DBCM by identifying primes by checking which derivative is the first without an edge between t and $t-1$. This way, the incremental approach of adding derivatives on a need-to-know basis as it is applied in DBCM learning could be fairly well evaluated. While not fully satisfying, I felt this provided a fair evaluation of how well finding prime variables worked for the DBCM algorithm. In total, there were four baseline algorithms, where two used the PC algorithm and two used the Bayesian algorithm. I will call them PC1, PC2, BAYES1, and BAYES2. Both PC approaches and both BAYES approaches were identical in the way they established the prime variables, but they differed in how the contemporaneous edges were found, which I will discuss now.

Once those latent differences were found, I used the PC algorithm and Bayesian algorithm to recover the contemporaneous structure, but without imposing the structure of a DBCM. In this step, PC1 and BAYES1 were identical (except for the used algorithm), and PC2 and BAYES2 were identical as well. For PC1 and BAYES1 the structure was checked of the variables in the second time slice of the learned network in the first step. This was compared to the true structure and statistics are reported below. For the PC2 and BAYES2 algorithms I used the prime variable information from the first step and then used a data set where each record in the data set is one time slice (just like with DBCM learning). The only difference is that I was not imposing the DBCM structure (e.g., allow edges between integral variables) and this will show how important it is to impose the DBCM structure.

In PC and DBCM I used a significance level of 0.01.The Bayesian approach starts with an empty network and then first greedily adds arcs using a Bayesian score with the K2 prior, and then greedily removes arcs. The Bayesian approach required discretizing the data for which I used 5 bins with approximately equal counts.

The results for the simple harmonic oscillator:

	# Derivs	# Too low	# Too high	# Edges	# Missing	# Extra	# Orientation
PC1	400	0	2	500	196	1161	130
PC2	400	0	2	500	499	104	1
BAYES1	400	66	288	500	299	1001	98
BAYES2	400	66	288	500	388	611	69
DBCM	400	0	2	500	2	6	3

The left part of the table shows the number of derivatives that had to be recovered over all runs (400) and how many of those derivatives were identified as too low or too high, compared to the true value. The right part of the table shows the number of edges that to be recovered (500) and then how many edges were missing, the number of extra edges, and the number of incorrectly oriented edges. The table below shows the results for the coupled harmonic oscillator:

	# Derivs	# Too low	# Too high	# Edges	# Missing	# Extra	# Orientation
PC1	800	0	99	1200	480	2368	278
PC2	800	0	99	1200	1002	295	169
BAYES1	800	0	741	1200	763	2024	102
BAYES2	800	0	741	1200	502	1735	257
DBCM	800	0	2	1200	7	16	77

The tables show that the method is effective at both learning the correct difference variables and in learning contemporaneous structure of these systems. For the simple harmonic oscillator, the PC baselines are performing exactly the same as DBCM, however, when the network gets more complicated, like the coupled harmonic oscillator, there is a clear difference. This implies that searching for prime variables by adding derivatives in an incremental approach is superior to adding them all at once. Also, in all cases the second step makes a big difference between baselines and DBCM, most likely because enforcing the DBCM structure is essential.

I did try other significance levels besides 0.01, but all results showed the same trend where DBCM learning clearly outperformed the baseline approaches. Lowering the significance value is a tradeoff between decreasing the number of extra edges, and increasing the number of missing edges.

I have to make some remarks about the edge orientations. Although the correct edges were found for different sets of parameters, getting the edge orientations right turned out to be more difficult. In particular, with most set of parameters there was almost no correlation between x_1 and a_1, which should not be the case. In this case, additional edge orientations were in the output of the DBCM learning algorithm, because of the unconditional independence of x_1 and a_1 instead of being conditional independent given F_{x1} and F_{x2}. It is apparent that the system being faithful is sensitive to the choice of parameters.

If noise is added to the integral equations, e.g., when the acceleration does not always exactly carry over to the velocity, results may be different. To investigate the influence of this noise, I generated data for the coupled harmonic oscillator that contained noise for the integral equations. Here is the table for normally distributed noise with standard deviation 0.01:

	# Derivs	# Too low	# Too high	# Edges	# Missing	# Extra	# Orientation
PC1	800	0	95	1200	397	2653	246
PC2	800	0	95	1200	1011	554	96
BAYES1	800	0	772	1200	679	1958	222
BAYES2	800	0	772	1200	295	1700	253
DBCM	800	0	15	1200	26	56	136

And here is the table for a standard deviation of 0.1:

	# Derivs	# Too low	# Too high	# Edges	# Missing	# Extra	# Orientation
PC1	800	0	81	1200	442	2386	311
PC2	800	0	81	1200	1087	563	66
BAYES1	800	0	689	1200	776	2111	108
BAYES2	800	0	689	1200	496	1610	268
DBCM	800	0	135	1200	427	319	202

It is apparent that the more noise is added, the worse the DBCM learner performs, because one of its assumptions is violated. In the second table, the PC approach is even better in finding the derivatives than DBCM learning. But, overall, DBCM learning it is still better in finding the correct edges.

Granger causality models and VAR models were computed for some of the simulated data for the coupled harmonic oscillator just to illustrate how uninformative these models are when the latent derivatives are unknown. Those results are shown in Figure 25 at the left side and the right side, respectively. The Granger graph is more difficult to interpret than the DBCM because of the presence of multiple double-headed edges indicating latent confounders. It was noted that the sole integral variables appeared in the Granger graph with reflexive

edges, which might lead to an alternative algorithm for finding prime variables. However, the Granger graph does not provide enough information to perform causal reasoning. The VAR model is also difficult to interpret, as it attempts to learn structure over time of an infinite-order Markov model. The graph of Figure 25 at the right hand side shows that variable x_1 has 65 parents spread out over time-lags from 1 to 100 (binned into groups of $\Delta t = 20$) at significance level of 0.05. Thus while VAR models might be useful for prediction, they provide little insight into the causality of DBCMs.

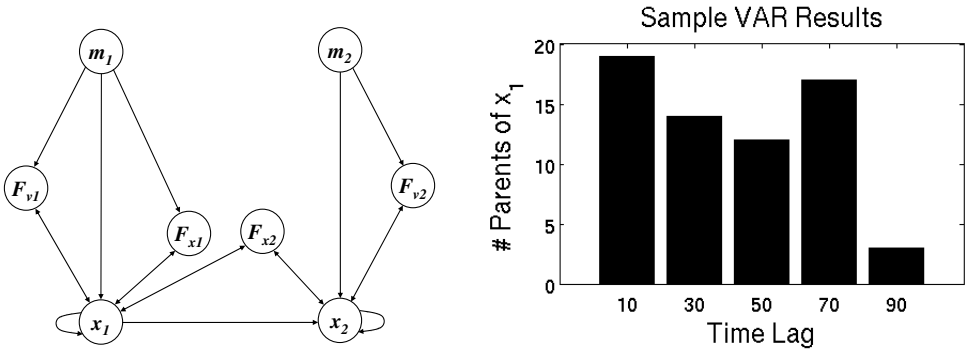

Figure 25: Left: A typical Granger causality graph recovered with simulated data. Right: The number of parents of x_1 over time-lag recovered from a VAR model (typical results).

I also performed a simple experiment with finding the causal structure of a nonlinear system, i.e., a system where the equations consist of nonlinear functions. In order to do so, a conditional independence test is required that works with any distribution. In Margaritis and Thrun [2001] an unconditional independence test is presented that works with any distribution, and the idea is the following. Suppose one wants to test if X is independent of Y, then these variables can be imagined as points on a 2-dimensional space. These points can be discretized in many different ways by imposing a grid onto this 2-dimensional space, for example, every midpoint between two points is a possible grid. Once we have defined a grid, it is possible to calculate the probability of independence by selecting which model is more likely; an independent one that is modeled by two separate multinomial distributions,

one for X and one for Y (requiring $\text{NumBins}(X) + \text{NumBins}(Y) - 2$ parameters), or a dependent one that is modeled as one multinomial distribution (requiring $\text{NumBins}(X) \times \text{NumBins}(Y) - 1$ parameters). This calculation, however, is for one fixed grid only and we have to take all possible grids into account. Because there are usually too many possible grids to average, instead new grid boundaries are added incrementally and the ones that have already been added are kept fixed. Each grid boundary is added in the position that increases the probability of dependence most, because if two variables are independent, they are independent for all resolutions.

In a follow-up paper, Margaritis [2005], the approach is extended to perform conditional independence tests as well. Suppose we want to test if X is independent of Y given Z. The idea is to sort the data along the Z "axis" and then recursively split the data set into 2 partitions along this dimension. In these partitions the joint distribution of X and Y should become more and more independent of Z. If the distribution is completely independent, then we can simply apply the unconditional test described before to test for conditional independence. The calculated probabilities for each of the partitions are then combined in a way that is explained in the paper.

I implemented the algorithm and it seemed to work fine for non-time-series data, although it is very slow. The reason is that for a conditional independence test involving three variables and 1,000 data points, one has to iterate through almost 1000,000,000 different grids for just one resolution. One way to resolve this would be to randomly select grids and calculate the probability of dependence, but I have not tried that. Applying this technique to time series data was less successful. I have tried learning the causal graph of a pendulum system, where the angular acceleration is a sine function of the angle. One conditional test of interest would be to calculate if the angular acceleration in two adjacent time steps is independent given the angle of the latter time step (this is simply a step in the DBCM algorithm). This test did not produce satisfactory results (it indicated conditional dependence instead of independence) and the reason seemed to be that the angle could not be partitioned in such a way that it became unconditionally independent of the angular acceleration in both of the time steps. It is not evident to me why this is the case and it is not easy to analyze what is happening because of the way the algorithm works. A more in-depth analysis is required. There has

also been recent work that take an approach based on kernel spaces [Tillman et al., 2009], which may lead to good results.

3.2 PREDICTIONS OF MANIPULATIONS

In this section I will show that DBCMs can be used to make predictions about manipulations, such as if a system will become unstable. I will use the example presented in Figure 26.

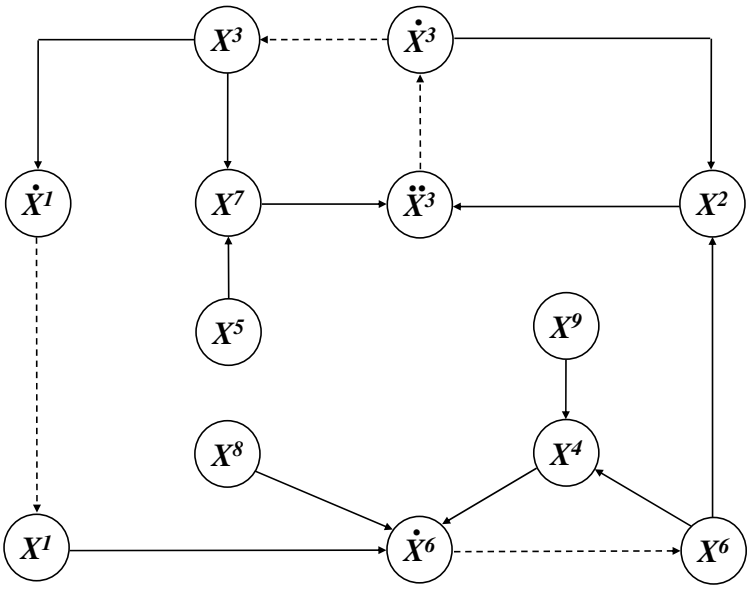

Figure 26: The DBCM graph I used to simulate data.

The aim is to relearn this DBCM from data that was generated and made available for the causality workbench competition. The input data[1] consisted of multiple time series that were generated first by parametrizing the model of Figure 26 with linear equations with independent Gaussian error terms, then by choosing different initial conditions for exogenous and dynamic variables and simulating 10,000 discrete time steps. As usual, the integral equations have no noise, because they involve a deterministic relationship.

[1] Downloadable from http://www.causality.inf.ethz.ch/repository.php?id=16

In a dynamic structure, different causal equilibrium models may exist over different time-scales. Which equilibrium models will be obtained over time are determined by the time-scales at which variables equilibrate. The causal structures are derived from the equations by applying the *Equilibrate* operator in the previous chapter [Iwasaki and Simon, 1994] and by assuming that at fast time-scales, the slower moving variables are relatively constant. In the example of Figure 26, the time-scales could be such that $\tau_6 \ll \tau_3 \ll \tau_1$, where τ_i is the time-scale of variable X_i, in which case, at time $t \sim \tau_6$ it would be safe to assume that X_3 and X_1 are approximately constant. Under these time-scale assumptions, Figure 27 shows the different (approximate) models that exist for the graph in Figure 26.

One obvious approach to learning the graph of Figure 26 (assuming no derivative variables are present in the data), is to try to learn an arbitrary-order dynamic Bayesian network, for example using the method of Friedman et al. [1998a]. However, this system is incorrect because it cannot represent the infinite order Markov chains that were discussed earlier. Another problem with learning an arbitrary Markov model to represent this dynamic system is that there are no constraints as to which variables may affect other variables across time, so in principle, the search space could be unneccessarily large.

The DBCM representation, on the other hand, implies specific rules for when variables can affect other variables in the future (when they instantaneously effect some derivative of the variable). Given that a derivative $\Delta^n X$ is being instantaneously caused, DBCMs also provide constraints on what variables can effect all $\Delta^i X$ for $i \neq n$.

After running the DBCM Learner on the data to obtain a causal structure, which resulted in the correct graph, I estimated the coefficients in the equations in order to be able to make quantitative predictions. I performed a multivariate linear regression for each variable on its parents and estimated the standard deviation of the noise term from the residuals. Now the task was to predict the effect of manipulating the variables. Each of the variables is manipulated once and the values of the first four time steps in the data set can be used to make predictions for time steps {5, 50, 100, 500, 1000, 2000, 4000, 10000}.

The results are shown in Figure 28, where the average Root Mean-Squared Error (RMSE) per time step for each manipulated variable is displayed. The graph shows that the error for the first few time steps is relatively small, but for all variables (except X_1) grows large in

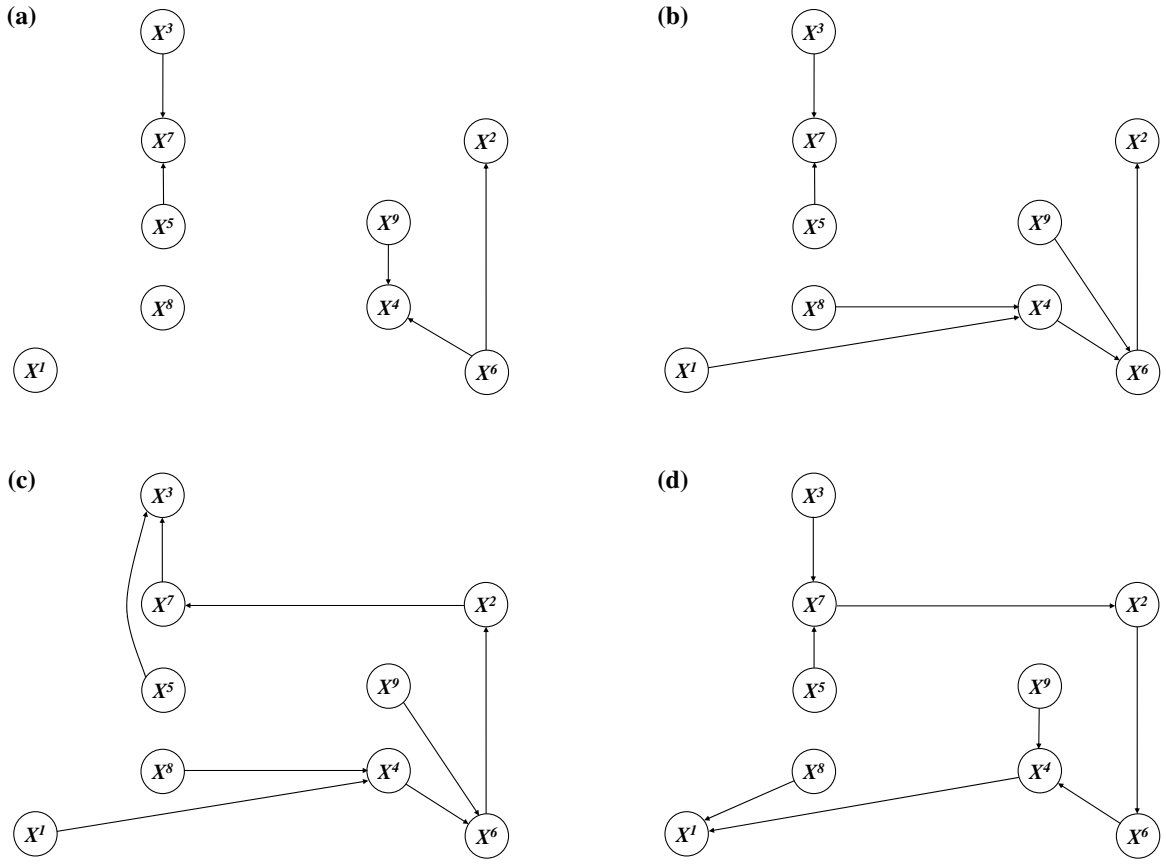

Figure 27: The different equilibrium models that exist in the sytem over time. (a) The independence constraints that hold when $t \sim 0$. (b) The independence constraints when $t \sim \tau_6$. (c) The independence constraints when $t \sim \tau_3$. (d) The independence constraints after all the variables are equilibrated, $t \gtrsim \tau_1$.

later times. Three variables in particular (X_2, X_7 and X_4) had astronomical errors in later times. These huge RMS errors are not indicative that the model was poor. In fact, since I generated the model, I could verify that the structure was exactly correct and the linear Gaussian parameters were very well identified. The reason for the unstable errors is that in the model of Figure 26, manipulating any variable except X_1 will approximately break the feedback loop of a dynamic variable and thus will in general result in an instability [Dash,

2003]. Feedback variable X_1 is a relatively slow process, so breaking this feedback loop does not have a large effect on the feedback loops of X_3 and X_6. Thus the absolute RMS error is expected to also be unstable for all manipulations but X_1, simply because such large values are predicted.

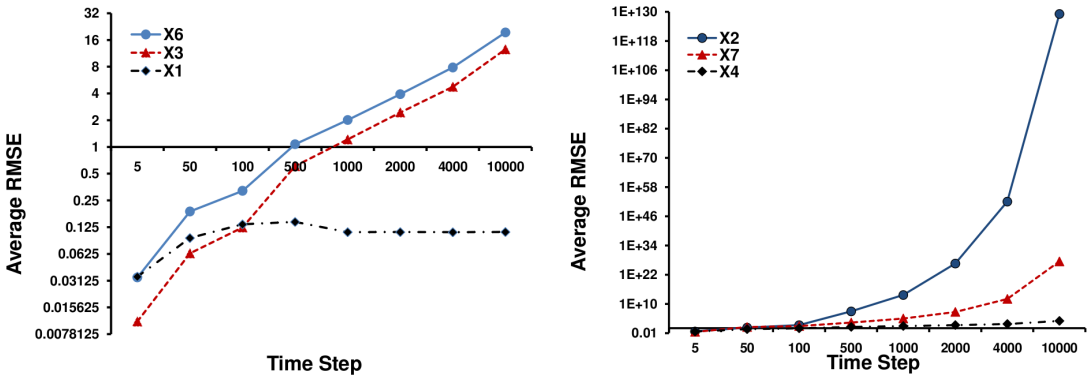

Figure 28: Average RMSE for each manipulated variable.

More important than getting the correct RMS error for these manipulations is the fact that the learned model correctly predicts that an instability will occur when any variable except X_1 is manipulated. In the absence of instability, the method has a very low RMS error, as indicated by the curve of variable X_1 in Figure 28. This fact is significant, because the model retrieved from the system when variable X_1 is allowed to come to equilibrium will not obey the EMC condition.

3.3 EEG BRAIN DATA

In this experiment, I attempted to learn a DBCM of the causal propagation of alpha waves, an 8-12 Hz signal that typically occurs in the human brain when the subject is in a waking state with eyes closed. Data was recorded by using electroencephalography (EEG), which records the electrical activity along the scalp produced by the firing of neurons within the brain.

Subjects were asked to close their eyes and then an EEG measurement was recorded. The data consisted of 10 subjects and for each subject a multivariate time series of 19 variables was recorded[2], containing over 200,000 time-slices at a sampling rate of 256 Hz. Each variable corresponds to a brain region using the standard 10-20 convention for placement of electrodes on the human scalp.

As an investigative approach, I first used the entire raw data set to determine what types of dynamic processes and causal interactions could be resolved. The significance was set to 0.01 and $k_{max} = 3$. The results for subject 10, which was typical are displayed in Figure 29 on the left. The circles represent the 19 variables that correspond to the brain regions. The top of this graph represents the front of the brain and the bottom the back, etc. The squares in the circle represent the derivatives that were found. The lower left is the original EEG signal, the lower right the first derivative, the top right the second derivative, and the top left the third derivative. In some regions, no derivatives were detected, so those squares have been left out.

These results were highly variable from subject to subject, but some commonalities persisted. First, most brain regions had at least one derivative, and the prime variables of the regions were highly connected in a fairly local manner. However, due to the high connectivity of this graph, it is difficult to get an understanding of what is happening. More quantitative analysis of these results is thus necessary. However, since the signals being measured are a superposition of several brain activities going on at once, a better approach might be to attempt to separate out specific activity and do a separate analysis for each one if possible.

Alpha rhythms are known to operate in a specific frequency band peaking at 10 Hz. To focus the results more on this process, I tried learning a DBCM using just the 10 Hz power signal over time. The data were divided into 0.5s segments, then a FFT was performed on that and the power was extracted of the 10 Hz bin for each time slice. When learning the DBCM, we used the same significance and k_{max} as before. The result for subject 10 is diplayed in Figure 29 on the right.

This graph shows a very different picture than the DBCM trained on all data. Here

[2]Data available at `http://www.causality.inf.ethz.ch/repository.php?id=17`

(and in typical subjects) there are only a few regions that required derivatives to explain their variation. The locations of those regions varied quite a bit from subject to subject, but there were some common patterns. Across all subjects, 16 of 20 occipital regions had at least one derivative present. This contrasts to frontal lobes where across all subjects only 1 of 70 frontal regions had one derivative or more. When a region had at least one derivative, rarely, if ever, did it also have an incoming edge from some region that did not have a derivative. This indicates that the regions containing the dynamic processes were the primary drivers of alpha-wave activity. Since most of these drivers occurred in the occipital lobes, this is consistent with the widely accepted view that alpha waves originate from the visual cortex.

There were many regions that did not require any derivatives to explain their signals. The alpha wave activity in these regions is very quickly ($< 0.5s$) determined given the state of the generating regions. One hypothesis to explain this is given by Gómez-Herrero et al. [2008] where they point out that conductivity of the skull can have significant impact on EEG readings by causing local signals to be a superposition of readings across the brain. Thus, if the readings of alpha waves detected in, say, the frontal region the brain is due merely to conductivity of the skull, we would effectively have instantaneous determination of the alpha signal in those regions given the value in the regions generating the alpha waves.

3.4 MEG BRAIN DATA

Magnetoencephalography (MEG) is an imaging technique used to measure the magnetic fields produced by electrical activity in the brain. In this experiment[3], subjects were asked to tap their left or right finger based on the instruction that appeared on a screen. 102 sensors were measuring the magnetic field, where each sensor consisted of three channels, namely two gradiometers and one magnometer. The gradiometers measure the magnetic field in two orthogonal directions along the scalp. The magnetometer measures the magnetic field in the

[3]I would like to thank the University of Pittsburgh Medical Center (UPMC), the Center for Advanced Brain Magnetic Source Imaging (CABMSI), and the Magnetic Resonance Research Center (MRRC) for providing the scanning time for the MEG data collection. I would also like to thank Dean Pomerleau and Gustavo Sudre for obtaining the data and making it available to me. The data is available upon request.

"Z-direction", i.e., perpendicular to the gradiometers. The sampling rate was 1000Hz.

Data for two subjects were available from -0.5s to 2s, where the stimulus indicating left or right was displayed at 0s. Typical reaction times were less than 0.5s, so all the relevant brain activity takes place in 0 to 0.5s. The subjects repeated this procedure more than hundred times so that for each finger at least 50 trials were available.

For finger tapping it is more or less known how the brain works. From the visual cortex in the occipital lobe of the brain a signal propagates to the motor cortex, which is located between the frontal lobe and parietal lobe. When the left finger is tapped, more activity should visible in the motor cortex of the right hemisphere of the brain, and vice versa.

Figure 30 shows the derivatives for DBCMs that were averaged over all right-tap trials where the data have been low-pass filtered to 100Hz and down sampled to 333Hz. Figure 31 shows the same for the left finger tap. It looks like the derivatives are generally higher in the visual cortex and motor cortex, just as one might expect. Similarly to the EEG data, these could be dynamic processes that quickly change and then instantaneously determine surrounding brain regions. It is not clear that a right finger tap results in a bigger increase in the left hemisphere and vice versa, and this could be because of several reasons. One reason is that only the samples in the range 0-0.5s were used, which were only about 125 points and more data may improve the results. Other reasons are that handedness may play a role; a right handed person usually has more activity in the left brain hemisphere and vice versa. The subject in the figures was right handed. Another thing that has to be taken into account is that DBCMs may not only capture activation, but also inhibition.

Next, I looked at the edges. For the same setting as described previously, I averaged all the edges over the different runs and plotted the ones that occurred more often than a certain threshold. The results are plotted in Figure 32. The results were surprising to me as the edges are somewhat similar to the iron filings lining up along a magnetic field, at least for the two gradiometers. The magnometer seems to be a combination of both gradiometers. This made me realize that it may be much better to combine the different measurement into one, and there is a methodology available called source localization that does exactly this. In fact, source localization converts back the measurements of the magnetic fields to the most likely electrical activity in the brain.

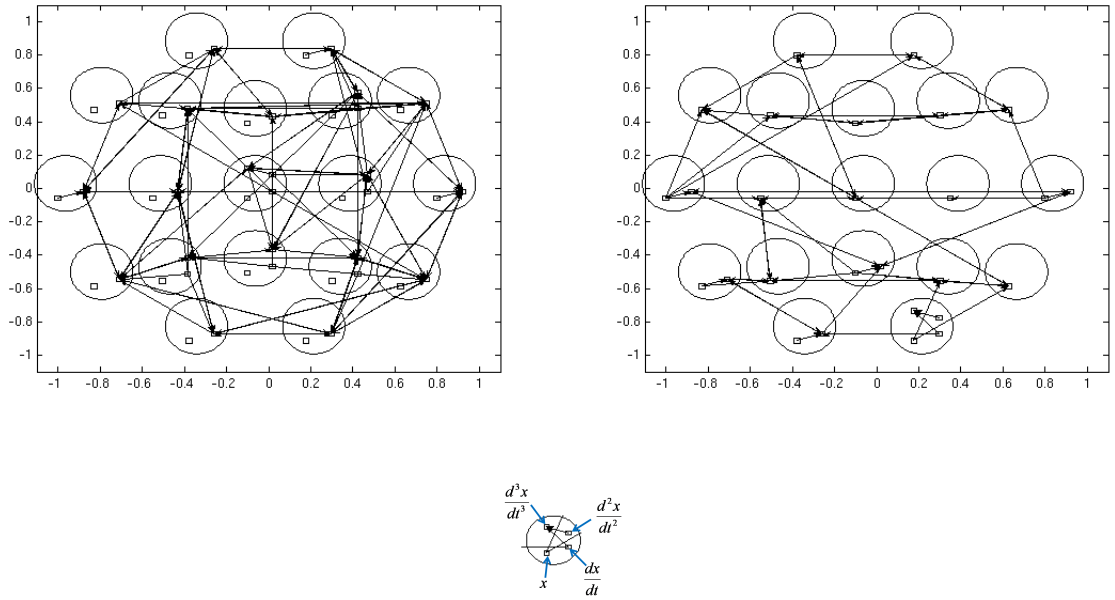

Figure 29: Left: Output after DBCM learning with the complete data. Right: Output after DBCM learning with the filtered data. Bottom: Legend of the derivatives.

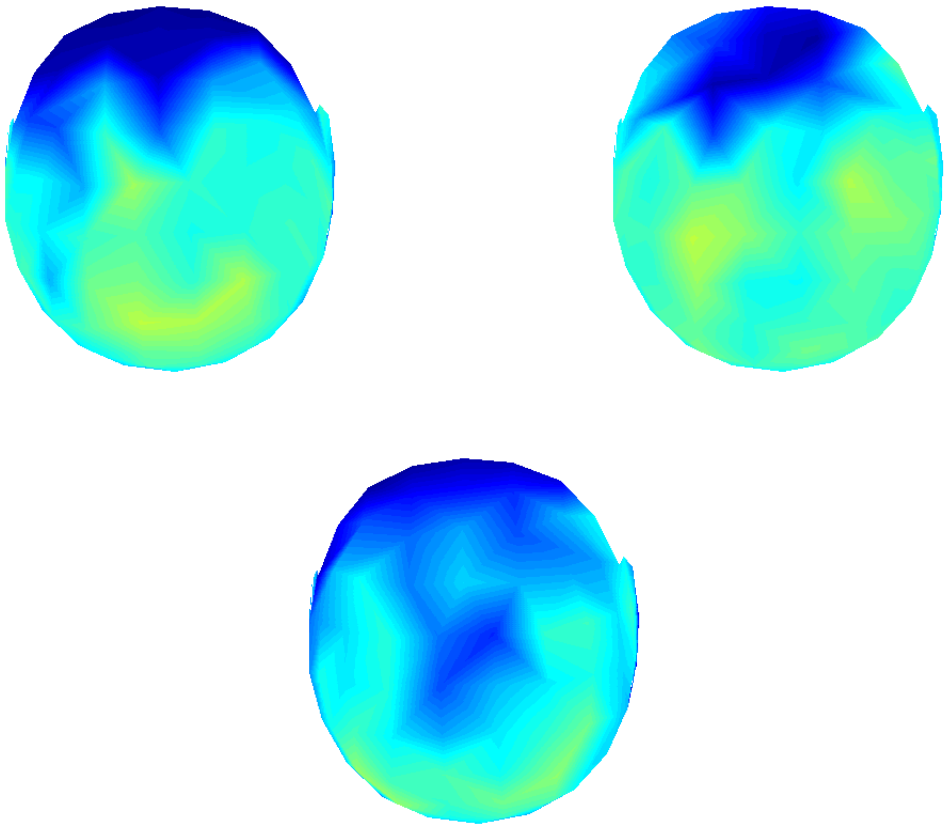

Figure 30: Right finger tap. Each image is a plot of the brain, where the top is the front. Blue means no derivative, green means first derivative, and yellow means second derivative. The top two images are the gradiometers and the bottom one is the magnetometer. It looks like the first gradiometer shows activity in the visual cortex, and the second one shows activity in the motor cortex.

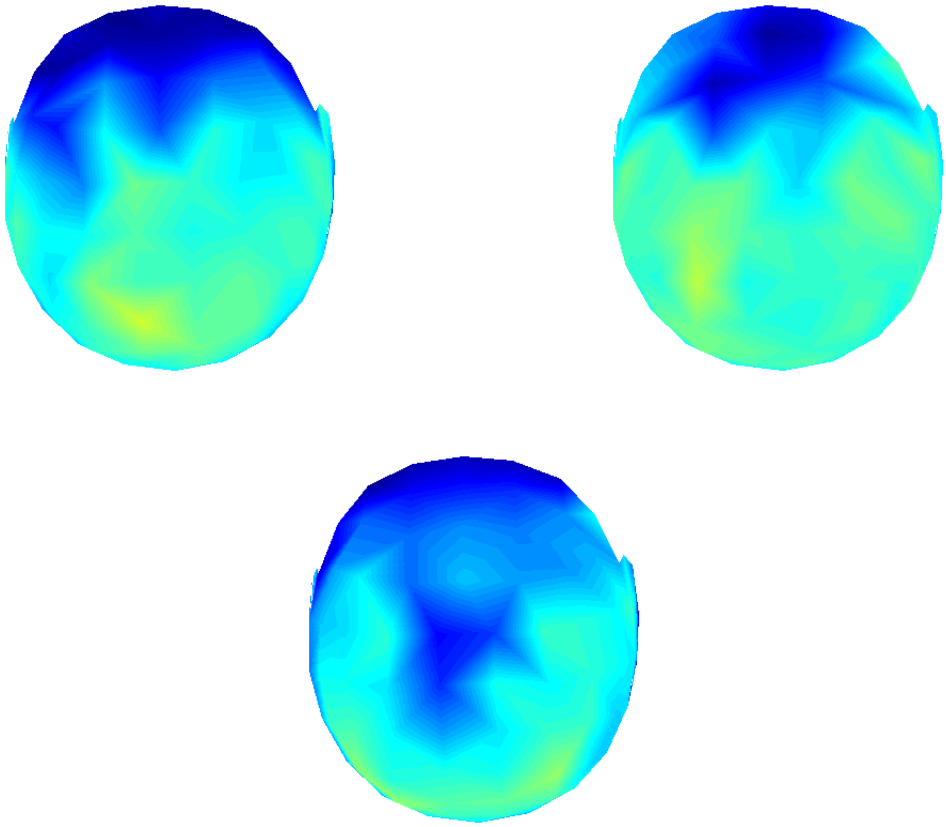

Figure 31: Left finger tap. This is somewhat similar to the right finger tap, but the derivatives are lower in general.

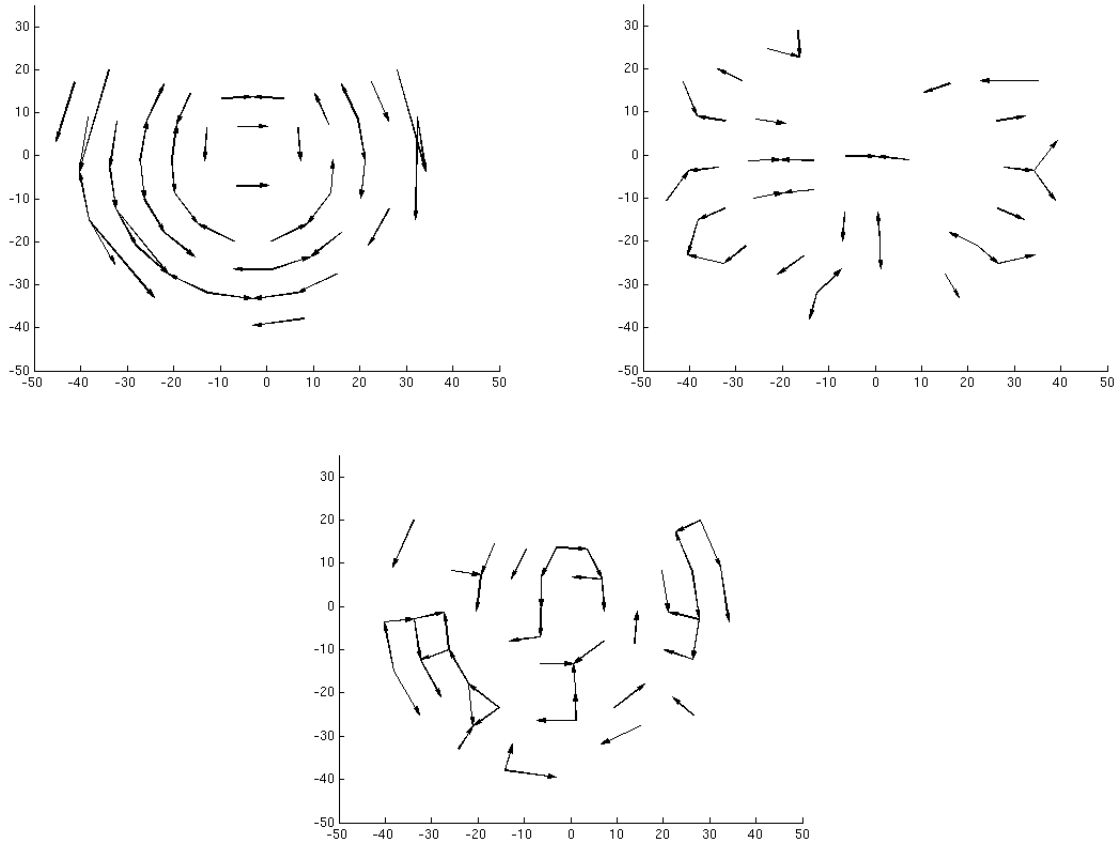

Figure 32: The two top figures show the edges for the gradiometers and the bottom one for the magnometer.

4.0 DISCUSSION

Recently, several philosophical and computational approaches to causality have used an interventionist framework to clarify the concept of causality [Spirtes et al., 2000, Pearl, 2000, Woodward, 2005]. The main feature of the interventionist approach is that causal models are potentially useful in predicting the effects of manipulations. One of the main motivations of such an undertaking comes from humans, who seem to create sophisticated mental causal models that they use to achieve their goals by manipulating the world.

Several algorithms have been developed to learn causal models from data that can be used to predict the effects of interventions [e.g., Spirtes et al., 2000]. However, Dash [2003, 2005] argued that when such equilibrium models do not satisfy what he calls the *Equilibration-Manipulation Commutability (EMC)* condition, causal reasoning with these models will be incorrect. This condition was explained in detail in Chapter 2. Because it is usually unknown whether EMC is satisfied, learning dynamic models becomes a necessity and that is the main motivation and goal of this dissertation. It was shown that existing approaches to learning dynamic models [e.g., Granger, 1969, Swanson and Granger, 1997] are unsatisfactory, because they do not perform a necessary search for hidden variables.

The main contribution of this dissertation is, to the best of my knowledge, the first provably correct learning algorithm called DBCM Learner that can discover dynamic causal models from data, which can then be used for causal reasoning even if the EMC condition is violated. As a representation for dynamic models I have used DBCMs, a representation of dynamic systems based on difference equations, inspired by the equations of motion governing all mechanical systems and based on Iwasaki and Simon [1994]. While there exist mathematical dynamic systems that can not be written as a DBCM, I believe that systems based differential equations are ubiquitous in nature, and, therefore, will be well approxi-

mated by DBCMs. Furthermore, because DBCMs are more restricted than arbitrary causal models over time, they can be learned much more efficiently and accurately.

I have shown that the DBCM Learner is capable of learning the correct model from simulated data from the harmonic oscillators. To the best in my knowledge, this is the first time that causal models can be learned for such mechanical systems. Furthermore, it was also empirically shown that DBCM learning can be used to predict the effect of manipulations, for example, if instabilities in a system will occur. I have argued that there is no existing representation available that is capable of learning a finite model of this and similar physical systems without first finding the correct latent derivative variables. This is because marginalizing out latent derivative variables results in an infinite-order Markov model. I have also shown that DBCMs can learn parsimonious representations for causal interactions of alpha waves in human brains that are consistent with previous research.

In general, I find it surprising that after nearly 50 years of developing theories for identification of causes in econometrics, and also the recent developments in causal discovery, that rarely, if ever, have researchers attempted to apply these theories to even the simplest dynamic physical systems. I feel my work thus exposes a glaring gap in causal discovery and representation, and I hope that by reversing that process—applying a representation that works well on known mechanical systems to more complicated biological, econometric and AI systems—we can make new inroads to causal understanding in these disciplines.

4.1 FUTURE WORK

First of all, one direction of future work would be to apply the DBCM Learner to more data sets. This may provide very useful insights into a variety of problems. If the results are not as good as expected, an analysis will have to be performed why this is the case and what assumptions are violated. This may also lead to the improvement of the DBCM Learner.

In its current form, the DBCM Learner is only capable of learning from continuous variables. It would be interesting to extend this with discrete variables that represent certain discrete events. It is not straightforward how to handle such cases. One way would be to

learn a separate DBCM for the data between two events to find out what effect the discrete events have on the causal relationships in the DBCMs.

One of the major problems with learning DBCMs is that none of the variables should be equilibrated, otherwise the learned model is susceptible to the problems associated with the EMC condition. Therefore, being able to automatically detect variables that equilibrate would help us to prevent learning incorrect models. In theory, detecting equilibration seems to be an easy problem, but developing an actual algorithm may turn out not to be easy.

Lastly, I will briefly mention a few other interesting issues. One of them involves the DBCM representation. DBCMs are a representation of difference (or differential) equations. However, besides differential equations, in nature also many phenomena are described accurately by partial differential equations. Supporting such equations may make DBCMs applicable to an even wider spectrum of problems. Another topic involves the time series that are used as input to the DBCM Learner. In this dissertation I have assumed that all time steps are uniform, however, certain data recordings may have non-uniform time steps and it may not be straightforward to apply DBCM learning to this data. Another, somewhat related issue, would be to use a more accurate way than simply taking differences to calculate values for the latent derivatives.

APPENDIX A

BAYESIAN NETWORKS

Bayesian networks can be seen as a marriage between graph theory and probability theory. They consist of a qualitative part in the form of a graph that encodes conditional independencies. This graph is enhanced with a quantitative part in the form of local probability distributions that together constitute a joint probability distribution over all the variables involved. I will only introduce important basic concepts, starting with the formal definition of a Bayesian network. For a more elaborate exposition, the reader is referred to an introductory text of Pearl [1988], for example.

Definition 20 (Bayesian network)**.** *A Bayesian network is a pair $\langle G, P \rangle$, where G is a directed acyclic graph (DAG) over a set of variables $\boldsymbol{X}$, and P is a joint probability distribution over $\boldsymbol{X}$ that can be written as*

$$P(X_1, \ldots, X_n) = \prod_i P(X_i | Pa(X_i)),$$

where $Pa(X_i)$ denotes the set of parents of X_i in G.

There are many important connections between the qualitative and quantitative aspects of Bayesian networks. The most fundamental connection is the local Markov condition.

Definition 21 (local Markov condition)**.** *A directed acyclic graph G over $\boldsymbol{X}$ and a probability distribution $P(\boldsymbol{X})$ satisfy the local Markov condition if and only if for every $X \in \boldsymbol{X}$, it holds that $(X \perp\!\!\!\perp \boldsymbol{NonDesc}(X) \mid \boldsymbol{Pa}(X))$, where $\boldsymbol{Pa}(X)$ denotes the parents of X in G and $\boldsymbol{NonDesc}(X)$ denotes the non-descendents of X in G.*

One of the most important concepts in Bayesian networks is conditional independence. There are two ways of establishing conditional independence in Bayesian networks. One could read conditional independence statements from the graph by using, for example, *d*-separation, which is defined below. The other is by inspecting the joint probability distribution. There is a strong connection between the two sets of independencies. Every Bayesian network structure has a set of joint probability distributions associated with it that factorizes according to the graph and every conditional independence that can be read from the graph also holds in P. Conversely, for every joint probability distribution there is a graph structure, such that a subset of the conditional independencies in the probability distribution hold in the graph.

Definition 22 (d-separation)**.** *Let* $\mathbf{X}$*,* $\mathbf{Y}$*, and* $\mathbf{Z}$ *be three disjoint sets of variables contained in a directed acyclic graph* G*.* $\mathbf{X}$ *is d-separated from* $\mathbf{Y}$ *given* $\mathbf{Z}$ *in* G*, if and only if for every undirected path* U *between one node in* $\mathbf{X}$ *and another node in* $\mathbf{Y}$ *at least one of the following two conditions hold:*

1. *There is a triplet* $A \rightarrow B \rightarrow C$ *or* $A \leftarrow B \rightarrow C$ *on* U *and* B *is in* $\mathbf{Z}$*.*
2. *There is a triplet* $A \rightarrow B \leftarrow C$ *on* U *and neither* B *nor any of its descendents are in* $\mathbf{Z}$*.*

A.1 CAUSAL BAYESIAN NETWORKS

In the preceding discussion there was no need to refer to causality, because formally Bayesian networks are just a compact representation of joint probability distributions. However, the directed arcs of the graphical structure of a Bayesian network can be given a causal interpretation and there is indeed a variant of Bayesian networks that are called causal Bayesian networks. This interpretation is formalized by the causal Markov assumption, which is defined analogously to the local Markov condition.

Definition 23 (causal Markov condition)**.** *A causal DAG* G *over* $\boldsymbol{X}$ *and a probability distribution* $P(\boldsymbol{X})$ *generated by the causal structure of* G *satisfy the causal Markov condition if and only if for every* $X \in \boldsymbol{X}$*, it holds that* $(X \perp\!\!\!\perp \boldsymbol{NonDesc}(X) \mid \boldsymbol{Pa}(X))$*, where* $\boldsymbol{Pa}(X)$

denotes the direct causes of X in G and $\boldsymbol{NonDesc}(X)$ *denotes the non-effects of X in G.*

A.2 LEARNING CAUSAL BAYESIAN NETWORKS

There are two main approaches to learning causal Bayesian networks, namely score-based search and constraint-based search. In the next two paragraphs, these approaches will be discussed.

A.2.1 Axioms

Spirtes et al. [2000] state three axioms for connecting probability distributions to causal models:

1. Causal Markov condition.
2. Causal minimality condition.
3. Faithfulness condition.

The causal Markov condition has been defined earlier in the previous section. Let $P(\boldsymbol{X})$ be a probability distribution over $\boldsymbol{X}$ and G be a graph over $\boldsymbol{X}$. Then the causal Markov condition is satisfied if and only if variable X is independent in $P(\boldsymbol{X})$ of all its non-effects in G given its direct causes in G.

The definition of the causal minimality condition is given next:

Definition 24 (causal minimality condition)**.** *Let $P(\boldsymbol{X})$ be a probability distribution over $\boldsymbol{X}$ and G be a graph over $\boldsymbol{X}$. Then $\langle G, P\rangle$ satisfies the causal minimality condition if and only if for every proper subgraph H of G over nodes $\boldsymbol{X}$, the causal Markov condition on the pair $\langle H, P\rangle$ is not satisfied.*

A fully connected graph always satisfies the causal Markov condition, because there are no conditional independencies implied by the graph. However, it does not satisfy the causal minimality condition if there is at least one (conditional) independence in the probability distribution. This is one simple example of a violation the causal minimality condition.

The faithfulness condition restricts the allowable connection between a graph G over $\boldsymbol{X}$ and a probability distribution P over $\boldsymbol{X}$ even more by requiring that all and only the conditional independencies of the causal Markov condition to the graph are also true in probability distribution P. Here is the formal definition:

Definition 25 (faithfulness condition)**.** *Let G be a causal graph and P a probability distribution generated by G. $\langle G, P\rangle$ satisfies the faithfulness condition if and only if every conditional independence relation true in P is entailed by the causal Markov condition applied to G.*

One of the consequences of this assumption is that deterministic relationships are not allowed, as they introduce conditional independencies not captured by the Markov condition. For example, suppose we have a causal graph $A \rightarrow B \rightarrow C$ then one conditional independence is implied, namely $(A \perp\!\!\!\perp C \mid B)$. However, if we assume that both arcs are deterministic relationships, then knowing either one of the three values will make the other two independent in the probability distribition, so $(A \perp\!\!\!\perp B \mid C)$ and $(B \perp\!\!\!\perp C \mid A)$ are also implied.

Another way the faithfulness condition can be violated is when there are several paths from variable A to B, but in such a way that the different paths of A to B cancel each other out completely, as if A would have no influence on B.

I will now discuss two different approaches to learning Bayesian networks.

A.2.2 Score-Based Search

Score-based learning algorithms search for the highest scoring Bayesian network given the data. Usually, a greedy search is combined with one of several different scoring functions that have been developed, such as the Bayesian Information Criterion (BIC) [Schwarz, 1978] and BDe [Cooper and Herskovits, 1992, Heckerman et al., 1995]. I will discuss both and assume complete data.

Both approaches combine the data likelihood $P(D|G, \Theta)$, where G is a graph, Θ the corresponding parameters, and D a data set, with a complexity penalty. A penalty is necessary, because a fully connected network will maximize the likelihood score. The BIC and BDe scores are both based on the posterior probability of the network structure. If G is a random variable representing all possible structures, then the posterior distribution is given by

$$P(G|D) \propto P(D|G)P(G) \ , \tag{A.1}$$

where $P(G)$ is the prior distribution over the different network structures, and $P(D|G)$ is known as the marginal likelihood and can be computed by marginalizing the corresponding network parameters:

$$P(D|G) = \int P(D|G,\Theta)P(\Theta|G)d\Theta \ ,$$

where $P(D|G,\Theta)$ is the data likelihood and $P(\Theta|G)$ is the prior distribution over the parameters, which could be hard to specify. This integral is hard to calculate in general, but under some simplifying assumptions sometimes even closed form solutions are attainable, as we will see later.

The BIC score circumvents calculating the exact marginal likelihood by looking at the asymptotic limit and, thus, ignoring the prior distribution over the parameters. This is justified if a large number of data points are available, because the prior distributions are becoming less influential as the data increases. A derivation of the asymptotic estimate by Schwarz [1978] results in the following equation for the log marginal likelihood:

$$\log P(D|G) = \log P(D|G,\hat{\Theta}_G) - \frac{\log N}{2}\,\mathrm{Dim}(G) + O(1) \ ,$$

where $\hat{\Theta}_G$ is the maximum likelihood estimate for network G, N the number of data records, $\mathrm{Dim}(G)$ is the dimension of the network calculated by counting the number of parameters in the network (the goal is to penalize complex structures), and $O(1)$ is a constant term that is independent of G and N. It is this equation that is used to score networks in the search.

The BDe score takes an alternative approach by making several assumptions so that the marginal likelihood can be calculated exactly. One such assumption is parameter independence. Let θ_{ij} denote the parameter vector of variable X_i having parent configuration Pa_i^j, N the number of samples, q_i the number of parent configurations of X_i, then parameter independence implies the following equation:

$$P(\theta|G) = \prod_{i}^{N} \prod_{j}^{q_i} P(\theta_{ij}|G) \ .$$

A final assumption is that the variables are multinomial and that the prior distribution over the parameters is given by a Dirichlet distribution. Given these assumptions, the marginal likelihood is calculated as (see Cooper and Herskovits [1992] for a derivation):

$$P(D|G) = \prod_{i}^{N} \prod_{j}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \ ,$$

where r_i are the number of possible values of X_i, $\alpha_{ij} = \sum_k^{r_i} \alpha_{ijk}$, and $N_{ij} = \sum_k^{r_i} N_{ijk}$.

Finally, we combine this result with Equation A.1 to calculate the total score for the structure. It is not necessary to calculate the exact posterior distribution, because normalizing has no effect on the ordering of the scores.

A.2.3 Constraint-Based Search

The PC algorithm requires four assumptions for the output to be correct. In the next four subsections each assumption is explained in detail. After that, the algorithm will be discussed.

A.2.3.1 Causal Sufficiency The set of observed variables should be causally sufficient. Causal sufficiency means that every common cause of two or more variables is contained in the data set. Causal sufficiency is a strong assumption, but there are algorithms that relax this assumption. The FCI algorithm [Spirtes et al., 2000] is capable of learning causal Bayesian networks without assuming causal sufficiency.

A.2.3.2 Samples From the Same Joint Distribution All records in the data set should be drawn from the same joint probability distribution. This assumption requires that all the causal relations hold for all units in the population. If the data are coming from two different distributions, it is always possible to introduce a node that acts as a switch between the two.

A.2.3.3 Correct Statistical Decisions Although the statistical decisions are not a part of the PC algorithm, their correctness is important for obtaining the conditional independence statements that are used as input to the PC algorithm. Therefore, the statistical decisions required by the algorithms should be correct for the population. This assumption is unnecessarily strong as even in the case of an incorrect outcome of a statistical test the PC algorithm may not be negatively influenced.

For discrete variables, a chi-squared test can be used to judge conditional independence. The continuous case is more complicated, because many different distributions are possible and tests are difficult to develop for the general case. Until recently, only the Z-test for multivariate normal distributions was widely used. Although this test also works in cases when there are deviations from a multivariate normal distribution [Voortman and Druzdzel, 2008], when the data are generated by nonlinear relationships, the test is likely to break down.

There are at least two lines of work that take an alternative approach by not assuming multivariate normal distributions at all. In Shimizu et al. [2005] the opposite assumption is made, namely that all error terms (except one) are non-normally distributed. This allows them to find the complete causal structure, while also assuming linearity and causal sufficiency, something that is not possible for normal error terms. Of course, this brings up the emperical question whether error terms are typically distributed normally or non-normally.

The second approach does not make any distributional assumptions at all. Margaritis [2005] describes an approach that is able to perform conditional independence tests on data that can have any distribution. However, the practical applicability of the algorithm is still an open question.

A.2.3.4 Faithfulness The probability distribution P over the observed variables should be faithful to a directed acyclic graph G of the causal structure. The precise definition of faithfulness was discussed earlier in this chapter. As mentioned before, one of the consequences of this assumption is that deterministic relationships are not allowed, as they introduce conditional independencies not captured by the Markov condition and also statistical tests do not work when there is no noise.

A.2.3.5 The Algorithm Constraint-based approaches take as input conditional independence statements obtained from statistical tests or experts, and then find a class of causal Bayesian networks that are implied by these conditional independencies. One prominent example of such an approach is the PC algorithm [Spirtes et al., 2000]. The PC algorithm works as follows:

1. Start with a complete undirected graph G with vertices $\mathbf{V}$.
2. For all ordered pairs $\langle X, Y \rangle$ that are adjacent in G, test if they are conditionally independent given a subset of $\mathbf{Adjacencies}(G, X) \setminus \{Y\}$. We increase the cardinality of the subsets incrementally, starting with the empty set. If the conditional independence test is positive, we remove the undirected link and set $\mathbf{Sepset}(X, Y)$ and $\mathbf{Sepset}(Y, X)$ to the conditioning variables that made X and Y conditionally independent.
3. For each triple of vertices X, Y, Z, such that the pairs $\{X, Y\}$ and $\{Y, Z\}$ are adjacent in G but $\{X, Z\}$ is not, orient $X - Y - Z$ as $X \rightarrow Y \leftarrow Z$ if and only if Y is not in $\mathbf{Sepset}(X, Z)$.
4. Orient the remaining edges in such a way that no new conditional independencies and no cycles are introduced. If an edge could still be directed in two ways, leave it undirected.

I illustrate the PC algorithm by means of a simple example (after Druzdzel and Glymour [1999]). Suppose we obtained a data set that is generated by the causal structure in Figure 33a, and we want to rediscover this causal structure. In Step (1), we start out with a complete undirected graph, shown in Figure 33b. In Step (2), we remove an edge when two variables are conditionally independent on a subset of adjacent variables. The graph in Figure 33 implies two (conditional) independencies, namely $(A \perp\!\!\!\perp B \mid \emptyset)$ and $(A \perp\!\!\!\perp D \mid \{B, C\})$, which leads to graphs in Figure 33c and 33d, respectively. Step (3) is crucial, since it is in this step where we orient the causal arcs. In our example, we have the triplet $A - C - B$ and C is not in $\mathbf{Sepset}(A, B)$, so we orient $A \rightarrow C$ and $B \rightarrow C$ in Figure 33e. In Step (4) we have to orient $C \rightarrow D$, otherwise $(A \perp\!\!\!\perp D \mid \{B, C\})$ would not hold, and $B \rightarrow D$ to prevent a cycle. Figure 33(f) shows the final result. In this example, we are able to rediscover the complete causal structure, although this is not possible in general.

The v-structures are responsible for the fact that learning the direction of causal arcs is possible. They will also become important in later chapters, so I will define them here:

Definition 26 (v-structure)**.** *Let X, Y, and Z be nodes in a Bayesian network. They form a v-structure on Y if and only if $X \to Y \leftarrow Z$ and no edge between X and Z.*

The defining characteristic of a v-structure is that it implies a different independence statement, compared to the other possible structures consisting of three nodes:

- $X \to Y \to Z$
- $X \leftarrow Y \leftarrow Z$
- $X \leftarrow Y \to Z$

While a v-structure implies $(X \perp\!\!\!\perp Z \mid \emptyset)$, the other three structures imply that $(X \perp\!\!\!\perp Z \mid Y)$. So if we find in the data a conditional independence statements such that $X - Y - Z$, and X and Z are independent unconditional on Y, we have identified a v-structure.

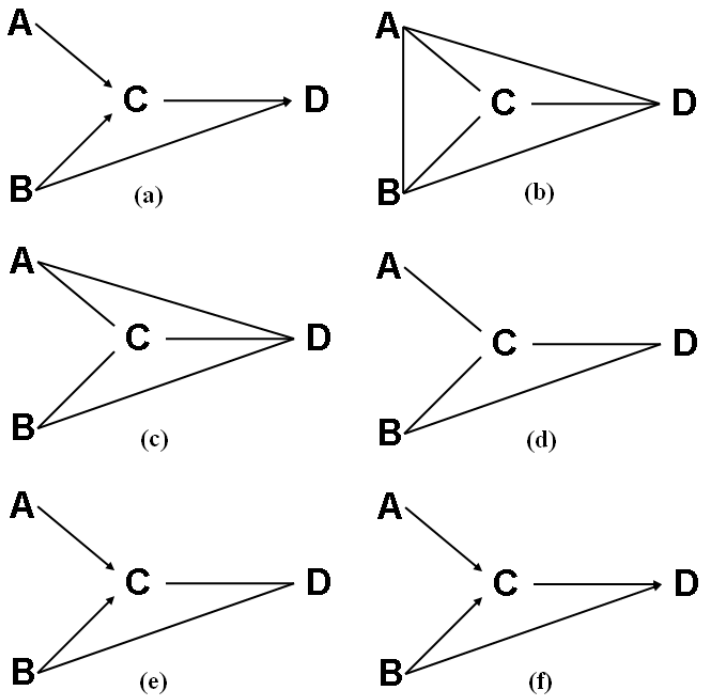

Figure 33: (a) The underlying directed acyclic graph. (b) The complete undirected graph. (c) Graph with zero order conditional independencies removed. (d) Graph with second order conditional independencies removed. (e) The partially rediscovered graph. (f) The fully rediscovered graph.

APPENDIX B

CAUSAL ORDERING

This section is based on Iwasaki [1988] and Iwasaki and Simon [1994], and is mainly included for self-containment. The causal ordering algorithm for three different kind of structures will be discussed: equilibrium, dynamic, and mixed structures. One subsection is devoted to each type of structure. I will now introduce an example that I will use throughout this section to illustrate the concepts. The example has been taken from Iwasaki and Simon [1994], but is slightly altered.

The example under consideration is a bathtub. Water is flowing into the tub with rate F_{in} and flowing out with rate F_{out}. The depth of the water is denoted by D, the pressure on the bottom of the tub is denoted by P, and the size of the valve opening is denoted by V.

- F_{in}, the input flow rate.
- D, the depth of the water in the tub.
- P, the pressure on the bottom of the tub.
- V, the size of the valve opening.
- F_{out}, the output flow rate.

This simple system is illustrated in Figure 34. Intuitively, the inflow rate of the water will have a causal effect on the depth of the water, which, in turn, determines the pressure on the bottom of the tub. The outflow rate is caused by the pressure and resticted by the size of the valve opening.

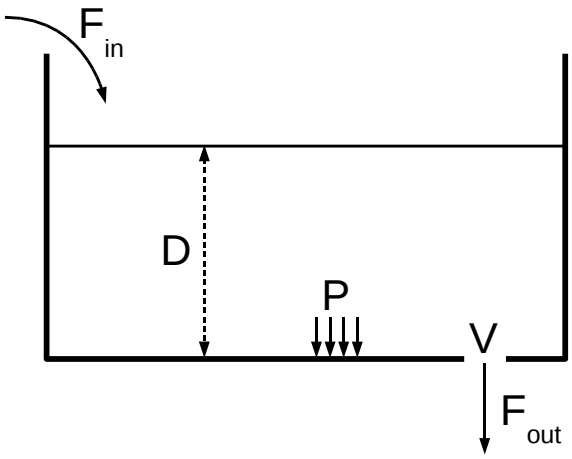

Figure 34: The bathtub example.

I will now look at the situations when the system is in equilibrium, when the system is dynamic, and when the system is in a mixed state.

B.1 EQUILIBRIUM STRUCTURES

The following definitions are taken from Iwasaki [1988] and Iwasaki and Simon [1994] and are only slightly altered for clarification and simplification. The causal ordering algorithm, described below, is a way of explicating the causal structure in a system of equations.

Definition 27 (self-contained equilibrium structure)**.** *A self-contained equilibrium structure is a system of n equilibrium equations in n variables that possesses the following special properties:*

1. *In any subset of k equations taken from the structure at least k different variables appear with non-zero coefficients in one or more of the equations of the subset.*
2. *In any subset of k equations in which m ($\leq k$) variables appear with non-zero coefficients,*

if the values of any $(m-k)$ variables are chosen arbitrarily, then the equations can be solved for unique values of the remaining k variables.

The first condition ensures that no part of the structure is overdetermined. The second condition ensures that the equations are not mutually dependent, because, if they are, the equations cannot be solved for unique values of the variables.

In case of the bathtub example, we have the following self-contained structure:

$f_1(F_{\text{in}})$	The input flow rate is a constant.	(B.1)
$f_2(D, P)$	The pressure is proportional to the depth of the water.	(B.2)
$f_3(V)$	The size of the valve opening is a constant.	(B.3)
$f_4(F_{\text{out}}, V, P)$	The outflow rate is proportional to the pressure.	(B.4)
$f_5(F_{\text{out}}, F_{\text{in}})$	In equilibrium, the inflow and outflow rate are equal.	(B.5)

Definition 28 (minimal self-contained subsets)**.** *The minimal self-contained subsets of an equilibrium structure are those subsets that do not themselves contain self-contained proper subsets.*

An example of a self-contained subset is Equation f_1.

Definition 29 (minimal complete subsets of zero order)**.** *Given a self-contained equilibrium structure, A, the minimal self-contained subsets of A are called the minimal complete subsets of zero order.*

There are two minimal complete subsets of zero order, namely Equation f_1 and f_3.

Definition 30 (derived structure)**.** *Given a self-contained equilibrium structure, A, and its minimal complete subsets of zero order, A', we can solve the equations of A' for the unique values of the variables in A', and substitute these values in the equations of $A - A'$. The structure, B, thus obtained is a self-contained equilibrium structure, and we call B a derived structure of first order. We can now find the minimal self-contained subsets of B, and repeat the process, obtaining the derived structure of second and higher order until the derived structure contains no proper complete subsets.*

The derived structure of first order consists of the following three equations, where the variables that are substituted are lowercase:

$$f_2(D, P) \tag{B.6}$$

$$f_4(F_{\text{out}}, v, P) \tag{B.7}$$

$$f_5(F_{\text{out}}, f_{\text{out}}) \tag{B.8}$$

Definition 31 (complete subsets of kth order)**.** *The minimal contained subsets of the derived structure of kth order will be called the complete subsets of kth order.*

Equation f_5 forms a complete subset of first order, and we can derive the derived structure of second order:

$$f_2(D, P) \tag{B.9}$$

$$f_4(f_{\text{out}}, v, P) \tag{B.10}$$

with f_4 as minimal complete subset of second order. The last step leaves us with the derived structure of third order having f_2 as minimal complete subset:

$$f_2(D, p)\ . \tag{B.11}$$

Definition 32 (exogenous and endogenous variables)**.** *If D is a complete subset of order k, and if a variable x_i appears in D but in no complete subset of order lower that k, then x_i is endogenous in the subset D. If x_i appears in D but also in some complete subset of order lower that k, then x_i is exogenous in the subset D.*

I will illustrate the concept of exogenous and endogenous variables by looking at Equation f_4, which contains the variables F_{out}, V, and P. Equation f_4 is a complete subset of second order. Variable F_{out} appears in another complete subset of lower than the second order, namely the first order, so F_{out} is exogenous with respect to the variables in f_4. Similarly, V is an exogenous variable. Variable P, however, is an endogenous variable relative to the variables in f_4, because it does not appear in a complete subset lower than order two.

Definition 33 (causal ordering in a self-contained equilibrium structure)**.** *Let β designate the set of variables endogenous to a complete subset B, and let γ designate the set endogenous to a complete subset C. Then the variables of γ are* directly causally dependent *on the variables of β (denoted as $\beta \rightarrow \gamma$), if at least one member of β appears as an exogenous variable in C. We can say also that the subset of equations B has* direct precedence *over the subset C.*

Let β be the endogenous variables in f_4, namely P. Let γ be the endogenous variables of f_2, namely D. Note that P is an endogenous variable in f_4 and an exogenous variable in f_2 and, by definition of causal ordering, $\beta \rightarrow \gamma$.

The resulting causal graph is shown in Figure 35. The result looks counterintuitive, because this is not how one would think about the causal processes in the bathtub. It would be intuitive to think that F_{in} would cause D, D causes P, and P and V cause F_{out}. However, it is important to realize that the causal graph is of the system in equilibrium. The relations in the graph hold when the system is in equilibrium, but not when it is disturbed from equilibrium, although they will hold again when the system returns to equilibrium. The way we should interpret the diagram, is as follows. In order for the system to be in equilibrium, F_{in} has to be equal to F_{out}. For F_{out} to be equal to F_{in}, P must have an appropriate value, which also depends on V. The value of D, in turn, is dependent on P. If we manipulate the value for V, then the system returns to a dynamic state and F_{out} increases. But when the system returns to equilibrium, the value of F_{out} must again be equal to F_{in}, and the manipulation of V will only change the values for P and D. The preceding interpretation of the equilibrium causal graph is a teleological explanation and it is not clear what the underlying mechanisms are. Therefore, I will now turn to the causal ordering algorithm in dynamic structures.

B.2 DYNAMIC STRUCTURES

The following first order differential equations are used to model the bathtub in a dynamic state:

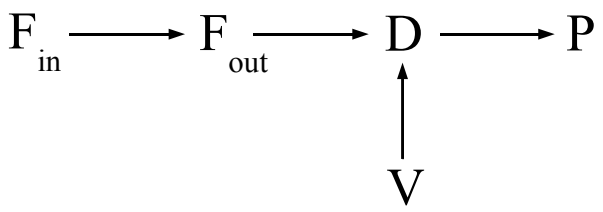

Figure 35: The equilibrium causal graph bathtub example.

$$\frac{dF_{\text{in}}}{dt} = c_1 \qquad \text{The rate of change of the inflow rate is exogenous.} \tag{B.12}$$

$$\frac{dD}{dt} = c_2(F_{\text{in}} - F_{\text{out}}) \qquad \text{The change in dept is determined by the inflow and outflow rate.} \tag{B.13}$$

$$\frac{dP}{dt} = c_4(D - c_5 P) \qquad \text{The change in pressure depends on the depth and pressure.} \tag{B.14}$$

$$\frac{dV}{dt} = c_6 \qquad \text{The size of the valve is exogenous.} \tag{B.15}$$

$$\frac{dF_{\text{out}}}{dt} = c_7(c_8 VP - F_{\text{out}}) \qquad \text{The rate of change in the inflow rate is exogenous.} \tag{B.16}$$

Analogous to a self-contained equilibrium structure, we can definine a self-contained dynamic structure.

Definition 34 (self-contained dynamic structure)**.** *A self-contained dynamic structure is a set of n first order differential equations involving n variables such that:*

1. *In any subset of k functions of the structure the first derivatives of at least k different variables appear.*

2. *In any subset of k functions in which r ($r \geq k$) first derivatives appear, if the values of any ($r - k$) first derivatives are chosen arbitrarily, then the remaining k are determined uniquely as functions of the n variables.*

The causal ordering algorithm for self-contained dynamic structures is easier than for equilibrium structures. By rewriting the equations into canonical form, i.e., by having only the derivative variables at the left side as is the case for the equations above, the causal structure is easily obtained. This is quite general, because every higher order equation can be rewritten as a system of first order equations. Every equation is considered to be a mechanism in the system and the derivative variables are caused by the variables in the right hand side of the equation. The causal graph is shown in Figure 36.

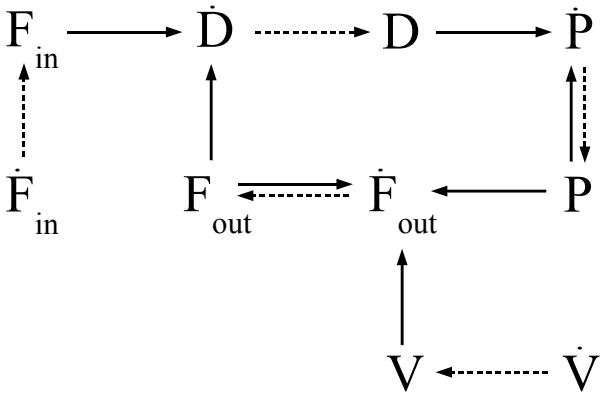

Figure 36: The dynamic causal graph bathtub example.

B.3 MIXED STRUCTURES

A mixed model is obtained from a dynamic model if one or more variables reach equilibrium.

Definition 35 (self-contained mixed structure)**.** *The set M of n equations in n variables is a self-contained mixed structure if and only if:*

1. *Zero or more of the n equations are first order differential equations and the rest are equilibrium equations.*
2. Inst(M) *is the set of instantaneous equations (no derivatives are present) and form a self-contained equilibrium structure when the variables and their derivatives are treated as distinct variables.*

A mixed model of the bathtub is obtained by equilibrating, for example, all the dynamic variables except $\frac{dD}{dt}$. The resulting causal graph is given in Figure 37. An arbitrary combination of variables that are equilibrated may result in a not self-contained mixed structure. An example is equilibriating all variables except F_{out}. In equilibrium, F_{in} and F_{out} are restored instantly, but if we look at the original causal structure in Figure 36, we see that the only causal path between the variables F_{in} and F_{out} runs through a lot of other variables. The variables on the path have to be equilibrated first, before F_{out} equilibrates.

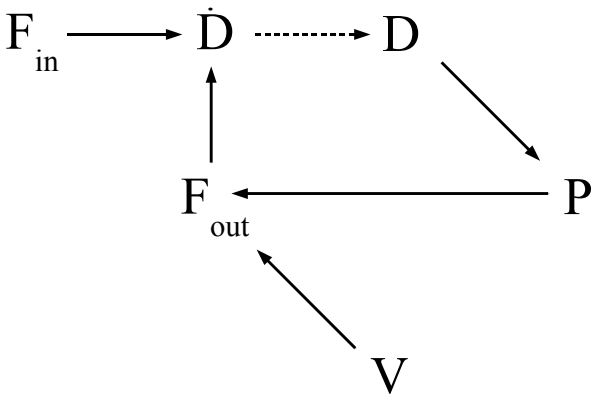

Figure 37: A mixed causal graph bathtub example.

APPENDIX C

PROOFS

This appendix contains the proofs for all the theorems in this dissertation. For the reader's convenience, the theorems are reprinted here.

Theorem 7 (detecting prime variables)**.** *Let* $\boldsymbol{V^t}$ *be a set of variables in a data set faithfully generated by a DBCM and let* $\boldsymbol{V^t_{all}} = \boldsymbol{V^t} \cup \Delta \boldsymbol{V^t}$*, where* $\Delta \boldsymbol{V^t}$ *is the set of all differences of* $\boldsymbol{V^t}$*. Then* $\Delta^j V_i^t \in \boldsymbol{V^t_{all}}$ *is a prime variable if and only if*

1. *There exists a set* $\boldsymbol{W} \subset \boldsymbol{V^t_{all}} \setminus V_i^t$ *such that* $(\Delta^j V_i^{t-1} \perp\!\!\!\perp \Delta^j V_i^t \mid \boldsymbol{W})$.
2. *There exists no set* $\boldsymbol{W'} \subset \boldsymbol{V^t_{all}} \setminus V_i^t$ *such that* $(\Delta^k V_i^{t-1} \perp\!\!\!\perp \Delta^k V_i^t \mid \boldsymbol{W'})$ *for* $k < j$.

Proof. $\Rightarrow$ If $\Delta^j V_i^t$ is a prime variable, then conditions 1 and 2 follow directly from the Markov condition.

$\Leftarrow$ Assume there exists a set $\boldsymbol{W}$ as stated in condition 1, and there exists no set $\boldsymbol{W'}$ as stated in condition 2. Because all $\Delta^n V_i^t$ are directly dependent on $\Delta^n V_i^{t-1}$, $n < k$, and since M is a DBC model, then by the faithfulness condition all $\Delta^n V_i^t$ are integral variables. The first variable $\Delta^n V_i^t$ that can be rendered independent of $\Delta^n V_i^{t-1}$ cannot itself be an integral variable and thus must be prime variable, which is in this case $\Delta^k V_i^t$. □

Theorem 8 (learning contemporaneous structure)**.** *Let* $\boldsymbol{V^t}$ *be a set of variables in a data set faithfully generated by a DBCM and let* $\boldsymbol{V^t_{all}} = \boldsymbol{V^t} \cup \Delta \boldsymbol{V^t}$*, where* $\Delta \boldsymbol{V^t}$ *is the set of all differences of* $\boldsymbol{V^t}$ *that are in the DBCM. Then there is an edge* $V_i^t - V_j^t$ *if and only if there is no set* $\boldsymbol{W} \in \boldsymbol{V^t_{all}} \setminus V_i^t, V_j^t$ *such that* $(V_i^t \perp\!\!\!\perp V_j^t \mid \boldsymbol{W})$.

Proof. $\Rightarrow$ Follows trivially from the Markov condition.

$\Leftarrow$ Assume there is no edge between V_1^t and V_2^t. Then there must exists a $\boldsymbol{V}^{t'} \subset \boldsymbol{V}^t \setminus \{V_1^t, V_2^t\}$ such that $(V_1^t \perp\!\!\!\perp V_2^t \mid \boldsymbol{V}^{t'})$. Because there is never an edge between two integral variables, we distinguish two cases. In the first case both variables are prime or static variables, and in the second case one of them is an integral variable. In the first case, we prove that V_1^t and V_2^t are independent conditioned on $\boldsymbol{Pa}(V_1^t) \cup \boldsymbol{Pa}(V_2^t)$. Please note that V_1^t and V_2^t are conditionally independent if the conditining set blocks all directed paths between V_1^t and V_2^t and contains no common descendents of V_1^t and V_2^t. If V_1^t is an ancestor of V_2^t the directed path is blocked by the parents of V_2^t and vice versa. It is impossible that any variable in the conditioning set is a common descendent, because a parent of V_1^t or V_2^t cannot at the same time be a descendent of V_1^t or V_2^t, respectively. This completes the first part of the proof. The second case is more complicated, because the parents of an integral variable are not included in $\boldsymbol{V}^t$. Let V_2^t be the integral variable and $\boldsymbol{V}_{\text{int}}^t$ be the set of all integral variables in time slice t. We construct a conditioning set $\boldsymbol{Pa}(V_1^t) \cup \boldsymbol{V}_{\text{int}}^t \setminus \{V_2^t\}$ that will d-separate V_1^t and V_2^t. Because integral variables have only outgoing arcs in the same time slice, we need only consider a directed path from V_2^t to V_1^t and a common cause of V_1^t and V_2^t in the previous time slice. A directed path from V_2^t to V_1^t is blocked by the parents of V_1^t. A common cause in the previous time slice is blocked by conditioning on all the integral variables, and this completes the proof. □

Theorem 9. *Let D be a DBCM with a variable X that has a prime variable $\Delta^m X$. The pdag returned by Algorithm 1 with a perfect independence oracle will have an edge between X and $\Delta^m X$ if and only if X is self-regulating.*

Proof. Follows by the correctness of the structure discovery algorithm (all adjacencies in the graph will be recovered) together with the definition of DBCMs (no contemporaneous edge can be oriented into an integral variable). □

Theorem 10. *Let G be the contemporaneous graph of a DBCM. Then for a variable X in G, $\boldsymbol{Fb}(X) = \emptyset$ if and only if for each undirected path P between X and $\Delta^m X$, there exists a v-structure $P_i \rightarrow P_j \leftarrow P_k$ in G such that $\{P_i, P_j, P_k\} \subset \boldsymbol{P}$.*

Proof. $\Rightarrow$ Assume $\boldsymbol{Fb}(X) = \emptyset$. Let P be an arbitrary path $P = P_0 \to P_1 - P_2 - \ldots - P_n - P_{n+1}$ with $P_0 = X$ and $P_{n+1} = \Delta^m X$, and let k be the number of cross-path colliders on that path. The path must have at least one (cross-path) collider, otherwise there will be a directed path from X to $\Delta^m X$ which contradicts the fact that $\boldsymbol{Fb}(X) = \emptyset$. If at least one of the cross-path colliders is unshielded the theorem is satisfied, so we only have to consider the case of shielded colliders. Now let $P_i \to P_j \leftarrow P_k$ be the first shielded cross-path collider (such that j is the smallest). We consider three cases:

1. $i < j < k$: There is a directed path from X to P_i since it is the first collider. Therefore, there can be no edge from P_k to P_i, because that would create a collider in P_i (and P_j would not be the first). So there must be an edge from P_i to P_k and this implies there is a directed path from X to P_k and we recurse and look for the first shielded cross-path collider after P_k.
2. $i, k < j$: Without loss of generality, there is a path $X \to \ldots \to P_i \to \ldots \to P_k \to \ldots \to P_j$, and edges $P_i \to P_j$, $P_k \to P_j$, and $P_i - P_k$. If $P_i \leftarrow P_k$ then there would be a collider in P_i which contradicts that P_j is the first one. Therefore, there must be an edge $P_i \to P_k$ and this implies there is a directed path from X to P_j and we recurse and find the first shielded cross-path collider after P_j.
3. $j < i, k$: Without loss of generality, there is a path $X \to \ldots \to P_j \ldots P_i \ldots P_k$, and edges $P_j \leftarrow P_i$ and $P_j \leftarrow P_k$. This results in two cross-path colliders in P_j. Now there are two possibilities, (a) they are both shielded which creates a directed path from X to P_k and we recurse like before, or (b) at least one cross-path collider is unshielded and resulting in the sought after v-structure.

Since there are only k cross-path colliders, case 1, 2, and 3a reduce the number of colliders towards zero. If there are no cross-path colliders left, there is a directed path from X to $\Delta^m X$ which contradicts our assumption that $\boldsymbol{Fb}(X) = \emptyset$. Therefore, eventually we must encounter case 3b and that proves one way of our theorem.

$\Leftarrow$ Assume all undirected paths between X and $\Delta^m X$ have such a v-structure. We prove by contradiction that there does not exist a directed path from X to $\Delta^m X$. Assume that $\boldsymbol{Fb}(X) \neq \emptyset$ and so there must be a path $P = X \to P_1 \to \ldots \to \Delta^m X$, and assume it

contains m such v-structures. Now let $P_i \to P_j \leftarrow P_k$ be the first v-structure (such that j is the smallest). We consider three cases:

1. $i > j$: There is a path $P_j \to \ldots \to P_i$ and also an edge $P_i \to P_j$ resulting in a cycle which is a contradiction.
2. $k > j$: Analogous to the first case.
3. $i, k < j$: Without loss of generality, assume that there is a path $X \to \ldots \to P_i \to \ldots \to P_k \to \ldots \to P_j \to$, and edges $P_i \to P_j$ and $P_k \to P_j$. So there is a directed path from X to P_j without a v-structure and we recurse to find the first v-structure after P_j.

Since there are only m cross-path colliders, eventually there will be a path with no colliders left. Since this path contains no v-structures, it contradicts the fact that all paths must have a v-structure and, therefore, $\boldsymbol{Fb}(X) = \emptyset$. □

BIBLIOGRAPHY

Tianjiao Chu and Clark Glymour. Search for additive nonlinear time series causal models. *Journal of Machine Learning Research*, 9:967–991, 2008. ISSN 1533-7928.

Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

Denver Dash. *Caveats for causal reasoning with equilibrium models*. PhD thesis, Intelligent Systems Program, University of Pittsburgh, Pittsbugh, PA, USA, April 2003. http://etd.library.pitt.edu/ETD/available/etd-05072003-102145/.

Denver Dash. Restructuring dynamic causal systems in equilibrium. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AIStats 2005)*. Society for Artificial Intelligence and Statistics, 2005. (Available electronically at http://www.gatsby.ucl.ac.uk/aistats/).

Selva Demiralp and Kevin Hoover. Searching for the causal structure of a vector autoregression. Working Papers 03-3, University of California at Davis, Department of Economics, March 2003.

Marek J. Druzdzel and Clark Glymour. Causal inferences from databases: Why universities lose students. In Clark Glymour and Gregory F. Cooper, editors, *Computation, Causation, and Discovery*, pages 521–539, Menlo Park, CA, 1999. AAAI Press.

Marek J. Druzdzel and Herbert A. Simon. Causality in bayesian belief networks. In *In Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 3–11. Morgan Kaufmann Publishers, Inc, 1993.

M. Eichler and V. Didelez. Causal reasoning in graphical time series models. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI-2007)*. Morgan Kaufmann, 2007.

Robert E. Engle and Clive W.J. Granger. Cointegration and error-correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, March 1987.

Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 139–147. Morgan Kaufmann, 1998a.

Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 139–147. Morgan Kaufmann, 1998b.

G. Gómez-Herrero, M. Atienza, K. Egiazarian, and J. L. Cantero. Measuring directional coupling between eeg sources. *NeuroImage*, 43(3):497–508, November 2008. ISSN 1095-9572.

Clive W.J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, July 1969.

C.W.J. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2(1):329–352, May 1980.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data, 1995.

David Hume. *A Treatise of Human Nature*. 1739.

Yumi Iwasaki. *Model based reasoning of device behavior with causal ordering*. PhD thesis, Pittsburgh, PA, USA, 1988.

Yumi Iwasaki and Herbert A. Simon. Causality and model abstraction. *Artificial Intelligence*, 67(1):143–194, May 1994.

Dimitris Margaritis. Distribution-free learning of Bayesian network structure in continuous domains. In *Proceedings of The Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

Dimitris Margaritis and Sebastian Thrun. A bayesian multiresolution independence test for continuous variables. In *UAI*, pages 346–353, 2001.

Alessio Moneta and Peter Spirtes. Graphical models for the identification of causal structures in multivariate time series models. In *JCIS*. Atlantis Press, 2006.

Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, New York, NY, USA, 2000.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

Judea Pearl and Thomas S. Verma. A theory of inferred causation. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *KR–91, Principles of Knowledge Representation and Reasoning:*

Proceedings of the Second International Conference, pages 441–452, Cambridge, MA, 1991. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Thomas Richardson and Peter Spirtes. Automated discovery of linear feedback models. In *Computation, Causation, and Discovery*, pages 253–302. AAAI Press, Menlo Park, CA, 1999.

Bertrand Russell. On the notion of cause. *Proceedings of the Aristotelian Society*, 13:1–26, 1913.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308:523–529, April 2005.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

Shohei Shimizu, Aapo Hyvarinen, Yutaka Kano, and Patrik O. Hoyer. Discovery of non-Gaussian linear causal models using ICA. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 525–53, Arlington, Virginia, 2005. AUAI Press.

Christopher A. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, January 1980.

Steven Sloman. *Causal Models: How People Think about the World and Its Alternatives*. Oxford University Press, USA, July 2005. ISBN 0195183118.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. Springer Verlag, New York, NY, USA, second edition, 2000.

Robert H. Strotz and H.O.A. Wold. Recursive vs. nonrecursive systems: An attempt at synthesis; Part I of a triptych on causal chain systems. *Econometrica*, 28(2):417–427, April 1960.

N.R. Swanson and C.W.J. Granger. Impulse response functions based on a causal approach to residual orthogonalization in vector autoregressions. *Journal of the American Statistical Association*, 92:357–367, January 1997.

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, *Advances in Neural Information Processing Systems 22*, pages 1847–1855. 2009.

Mark Voortman and Marek J. Druzdzel. Insensitivity of constraint-based causal discovery algorithms to violations of the assumption of multivariate normality. In *FLAIRS Conference*, pages 690–695, 2008.

James Woodward. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, USA, October 2005. ISBN 0195189531.