Information-Theoretic Causal Discovery

A dissertation submitted towards the degree
Doctor of Natural Sciences (Dr. rer. nat.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by Alexander Marx

Saarbrücken, 2021

Day of Colloquium	29 June 2021
Dean of Faculty	Prof. Dr. Thomas Schuster
Examination Committee	
Chair	Prof. Dr. Isabel Valera
Reviewer	Prof. Dr. Jilles Vreeken
Reviewer	Prof. Dr. Gerhard Weikum
Reviewer	Dr. Thijs van Ommen
Academic Assistant	Dr. Michael Kamp

Abstract

It is well-known that correlation does not equal causation, but how can we infer causal relations from data? Causal discovery tries to answer precisely this question by rigorously analyzing under which assumptions it is feasible to infer causal networks from passively collected, so-called observational data. Particularly, causal discovery aims to infer a directed graph among a set of observed random variables under assumptions which are as realistic as possible.

A key assumption in causal discovery is faithfulness. That is, we assume that separations in the true graph imply independencies in the distribution and vice versa. If faithfulness holds and we have access to a perfect independence oracle, traditional causal discovery approaches can infer the Markov equivalence class of the true causal graph—i.e., infer the correct undirected network and even some of the edge directions. In a real-world setting, faithfulness may be violated, however, and neither do we have access to such an independence oracle. Beyond that, we are interested in inferring the complete DAG structure and not just the Markov equivalence class. To circumvent or at least alleviate these limitations, we take an information-theoretic approach.

In the first part of this thesis, we consider violations of faithfulness that can be induced by exclusive or relations or cancelling paths, and develop a weaker faithfulness assumption, called 2-adjacency faithfulness, to detect some of these mechanisms. Further, we analyze under which conditions it is possible to infer the correct DAG structure even if such violations occur.

In the second part, we focus on independence testing via conditional mutual information (CMI). CMI is an information-theoretic measure of dependence based on Shannon entropy. We first suggest estimating CMI for discrete variables via normalized maximum likelihood instead of the plug-in maximum likelihood estimator that tends to overestimate dependencies. On top of that, we show that CMI can be consistently estimated for discrete-continuous mixture random variables by simply discretizing the continuous parts of each variable.

Last, we consider the problem of distinguishing the two Markov equivalent graphs $X \to Y$ and $Y \to X$, which is a necessary step towards discovering all edge directions. To solve this problem, it is inevitable to make assumptions about the generating mechanism. We build upon the idea which states that the cause is algorithmically independent of its mechanism. We propose two methods to approximate this postulate via the Minimum Description Length (MDL) principle: one for univariate numeric data and one for multivariate mixed-type data. Finally, we combine insights from our MDL-based approach and regression-based methods with strong guarantees and show we can identify cause and effect via L_0-regularized regression.

Zusammenfassung

Es ist bekannt, dass Korrelation nicht gleich Kausalität ist. Doch wie können wir kausale Zusammenhänge aus Daten schlussfolgern? Causal Discovery versucht exakt dieses Problem zu lösen, indem genau spezifiziert wird, unter welchen Annahmen kausale Netzwerke aus passiv gesammelten Daten, sogenannten Beobachtungsdaten, hergeleitet werden können. Ziel ist es, unter realistischen Annahmen, gerichtete Graphen aus Beobachtungsdaten zu inferieren.

Die Faithfulness Annahme ist essenziell für Causal Discovery. Diese nimmt an, dass Unabhängigkeiten in der Wahrscheinlichkeitsverteilung in direktem Zusammenhang mit Separationen im kausalen Graph stehen. Klassische Causal Discovery Algorithmen können unter dieser Annahme und mit Hilfe eines perfekten Unabhängigkeitstests den kausalen Graphen bis zu seiner Markov Äquivalenzklasse bestimmen. Diese besteht aus dem korrekten ungerichteten Graphen und einer Teilmenge der Kantenrichtungen. In der Praxis könnte allerdings weder die Faithfulness Annahme gelten, noch verfügen wir über einen perfekten Unabhängigkeitstest. Zudem ist es unser Ziel, alle Kantenrichtungen zu bestimmen und nicht nur die Markov Äquivalenzklasse des Graphen. Um diese Probleme zu überwinden oder zumindest abzuschwächen, verfolgen wir einen informationstheoretischen Ansatz.

Im ersten Teil dieser Dissertation betrachten wir kausale Mechanismen, welche die Faithfulness Annahme verletzen. Hierzu definieren wir eine schwächere Faithfulness Annahme, die es uns erlaubt einen Teil dieser Interaktionen zu finden. Zusätzlich analysieren wir unter welchen Bedingungen es möglich ist, die Kantenrichtungen eines solchen Mechanismus zu bestimmen.

Der zweite Teil setzt sich mit dem Testen von Unabhängigkeiten auseinander. Dazu stützen wir uns auf die konditionelle Transinformation (CMI); ein informationstheoretisches Maß zum Beziffern von Abhängigkeiten, basierend auf der Shannon Entropie. Wir entwickeln zwei konsistente Schätzer für CMI, einen spezifisch für diskrete Daten und einen generelleren Schätzer für diskrete, kontinuierliche und gemischte Daten. Um CMI auf gemischten Daten zu berechnen, diskretisieren wir zuerst die kontinuierlichen Datenpunkte mittels adaptiver Histogramme und arbeiten anschließend mit den diskretisierten Daten.

Im dritten Teil beschäftigen wir uns damit, die beiden Markov äquivalenten Graphen $X \rightarrow Y$ und $Y \rightarrow X$ zu unterscheiden. Dazu müssen wir Annahmen über den kausalen Mechanismus treffen. Wir bauen in unserem Ansatz auf das Postulat, das eine algorithmische Unabhängigkeit zwischen Ursache und Mechanismus—definiert durch Kolmogorov Komplexität—annimmt. Wir approximieren diese Inferenzregel mit Hilfe des Minimum Description Length Prinzips und entwickeln einen Ansatz für univariate numerische Paare und einen weiteren Ansatz für multivariate gemischte Daten. Zusätzlich zeigen wir, dass es unter relativ schwachen Annahmen möglich ist, Ursache und Wirkung mit Hilfe von L_0-regularisierten Regressionsmethoden zu bestimmen.

Für meine Eltern und Großeltern, die mich immer unterstützt haben.

Acknowledgments

First and foremost, I would like to say “Bedankt!” to my advisor Jilles Vreeken. I consider myself very fortunate to have Jilles as my advisor, who did not only provide me with countless valuable feedback and academic advice, but with whom I also share lots of enjoyable memories such as many joint conference trips, discussions about Saarland and the Dutch culture, as well as, intense but rewarding deadlines.

Next, I would like to thank Gerhard Weikum without whom this dissertation would not have been possible. Throughout my thesis, Gerhard was always supportive, provided me with valuable feedback, and last but not least, he supported my research visit to Amsterdam.

Connected to the latter, I would like to thank Joris Mooij for hosting my research visit. It was a pleasure to visit his research group at the University of Amsterdam and work together on a joint project.

A few of these chapters would also not exist without my co-authors Arthur Gretton, Joris Mooij, Lincen Yang and Matthijs van Leeuwen, whom I would like to thank for their valuable contributions and feedback to this work. I am proud to include both of these projects in my thesis and hope to continue our collaboration in the future.

I would like to thank the examination committee, consisting of Isabel Valera, Jilles Vreeken, Gerhard Weikum, Thijs van Ommen and Michael Kamp for carefully reading, considering and approving my thesis.

Thanks to all my colleagues at the EDA group! It was a joy to work together, discuss, have coffee breaks, travel, play card games, go bouldering, and much more.

Es ist mir wichtig, mich auch bei meinen Freunden und ganz besonders bei meiner Familie zu bedanken, die mich auf dem Weg zu dieser Dissertation durch alle Höhen und Tiefen begleitet haben. Danke!

Table of Contents

Chapter 1

Introduction

The urge to think about cause and effect is deeply embedded in human reasoning. Let us imagine we walk down a busy shopping street on a sunny afternoon. When we briefly walk past a kid enjoying some ice cream, we notice that the ice cream has almost melted. Immediately, our brain starts to think about reasons that could have caused the ice cream to melt. Since it is not that hot outside, we quickly figure out that the child probably did not eat the ice cream very fast. To come to this conclusion, we make use of the causal model that we have in our heads. We know that ice cream typically melts when it is warmer than zero degrees celsius and that depending on how hot it is, it will take more or less time for it to melt. Combining these two aspects, we come to the conclusion that to this day, it must have taken some time for the ice cream to melt that much. Using our causal knowledge on this matter, we can also try to make predictions. For example, will the ice cream melt completely if the child continues eating at this pace? The main emphasis here is that we need to have a model in our minds to answer such questions.

But what if we do not have such a model or knowledge for a specific question, for example, does gene A influence or cause gene B? Although one might argue that a gene does not *cause* another gene, we will not go into a philosophical discussion on the meaning of the word cause and call a relationship causal if it has a direction attached to it. To figure out if gene A causes gene B, we could try to run a controlled experiment and knock out gene A. Simply put, A would not be available in the system or unable to function. If we then observe that gene B is not produced anymore, we could trace back that this had something to do with knocking out A. To verify that gene A influences gene B and not vice versa, we could repeat the experiment but knock out B

instead of A. Suppose this does not change the expression of gene A. In that case, we can conclude a causal effect from gene A to gene B, which we denote as $A \to B$. Doing such controlled experiments is, however, not always an option because there might be ethical problems, e.g. in psychological studies, or such experiments might be too expensive, or they are simply too difficult to conduct. On the flip side, we often have access to passively collected data, so called observational data. Extracting causal knowledge from such observational data is often referred to as causal discovery (Spirtes et al., 2000). In the following, we will first briefly explain the most fundamental concepts and assumptions of causal discovery at a toy example. Subsequently, we point out open problems and shortcomings of some of these assumptions and formulate the research questions that we investigate in this thesis.

1.1 A Gentle Introduction to Causal Discovery

The goal of causal discovery is to infer a causal network among a set of attributes or features, e.g. a set of genes, from observational data. In a causal network, each node represents a feature from the data set and a directed edge between two nodes X and Y denotes a causal relationship, e.g. $X \to Y$, denotes that X is the cause of Y. Without making any assumptions, observational data is, however, not sufficient to unambiguously infer causal relationships, as opposed to data gathered through controlled experiments (Pearl, 2009). Yet, if we are willing to make some assumptions, e.g., assume that features that are not even correlated are also not causally related, we can infer causal relationships from observational data (Spirtes et al., 2000). Ideally, we would prefer to make these assumptions as light-weight as possible such that they are likely to hold in practice. Below, we will provide a detailed example of a causal network, which we will use to explain the main concepts and assumptions of causal discovery.[1]

Consider the causal network in Figure 1.1, which describes an office environment. We have Jack and Jill, who share the office and should, due to a global pandemic, come to the office according to a certain office plan (*Plan*). In addition, Jack and Jill have an office plant. We can model how much this plant grows (*Growth*) dependent on its exposure to sunlight (*Light*) and dependent on the amount of water it has access to (*Water*). In this scenario, *Light* and *Water* are both direct causes or parents of *Growth*, which is encoded in the graph via the two directed edges pointing towards *Growth*. Sunlight is an independent factor that has no parents in this network. In contrast, the amount of water the plant gets, strongly depends on whether or not Jack and Jill are

[1] This introduction is mostly tailored towards constraint-based causal discovery, which relies on conditional independence tests. However, the main ideas do also translate to other classes of algorithms, such as score-based methods.

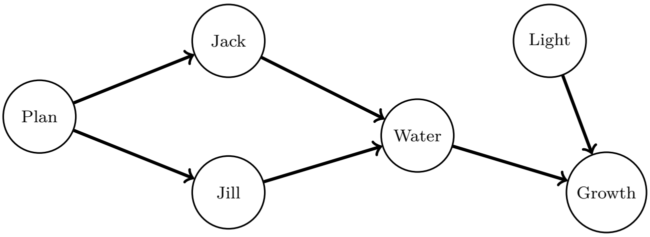


Figure 1.1: Office Plant Network: The above toy network describes factors that causally influence the growth of an office plant in a shared office. Jack and Jill share the office and the corresponding nodes represent their presence at the office. Both Jack and Jill are supposed follow an office plan (*Plan*), which is generated by a certain mechanism. The presence of Jack and Jill influences the amount of water the plant gets (*Water*). Together, the amount of water and the exposure to sunlight (*Light*) determine the plants growth (*Growth*) through some complicated mechanism.

at the office, which in turn depends on the office plan. Note, however, that the office plan is no direct cause or parent of *Water*, but can be referred to as an ancestor of *Water*. Vice versa, we call *Water* a descendent of *Plan*, *Jack* and *Jill*, whereas only for *Jack* and *Jill*, *Water* is also a child node in the graph.

At a high level, such a causal network is an accessible way to summarize the more complicated causal mechanism itself. Similar to the child with the ice cream, for which we not only knew that heat melts the ice cream but also had a rough mechanism in our minds how those quantities relate, the plant growth also follows a complicated mechanism dependent on the amount of water available and its exposure to sunlight. In the real world, there also exist other factors that influence the growth of a plant, e.g., small fluctuations of the temperature in the room, the humidity and many more. In our model, we abstract those away as noise, and model the plant growth as

$$Growth := f(Light, Water, N) ,$$

where f is a complicated function of *Light*, *Water* and an independent noise variable N, which could, for example, be modelled as Gaussian distributed. We also call N an exogenous or unobserved variable. In a similar way, we can describe each node in the network

$$\begin{aligned} Plan &:= f_1(N_1) & Jill &:= f_4(Plan, N_4) \\ Light &:= f_2(N_2) & Water &:= f_5(Jack, Jill, N_5) \\ Jack &:= f_3(Plan, N_3) & Growth &:= f_6(Light, Water, N_6) . \end{aligned}$$

Since both *Plan* and *Light* do not have any parent nodes, they purely depend on external nodes. Altogether, such a set of equations allows us to model each node as a random variable and define a joint distribution P among them. As can be seen from the above equations, we use different noise variables for each observed variable. A quite common assumption that we also make throughout this thesis is to assume that all external variables are independent of each other. Thus, no unobserved variable influences more than one of the observed variables. This assumption is called causal sufficiency, i.e., we assume that we observe all relevant variables. Besides, we will assume that all causal relationships are acyclic. There also exist approaches that aim to relax these assumptions (Spirtes et al., 2000; Forré and Mooij, 2019), but we do not focus on these in this thesis.

Under the assumption that causal sufficiency holds and there are no cyclic causal relations, the true causal network can be described by a directed acyclic graph (DAG), similar to the one shown in Figure 1.1. To infer such a network only given the distribution P or a large data set that we assume to be drawn independent and identically distributed (i.i.d.) according to P, we need to make assumptions that hold for the network and the distribution. Most common are the causal Markov condition (CMC) and the faithfulness assumption (Spirtes et al., 2000). Together, they state that each independence statement in the distribution is due to a separation in the true graph and vice versa. As a simple example, consider *Plan* and *Light* in Figure 1.1. Clearly, both variables are generated from two independent noise variables N_1 and N_2 and are thus independent of each other. Hence, we correctly conclude from the faithfulness assumption that both nodes are not adjacent in the true causal graph. From the graph, on the other hand, we see that all directed paths from *Jill* to *Growth* are blocked by the node *Water*. Accordingly, we expect that due to the causal Markov condition *Jill* will be independent of *Growth* if we condition on *Water*, denoted as $\mathit{Jill} \perp\!\!\!\perp \mathit{Growth} \mid \mathit{Water}$. Since both nodes are connected through a directed path in a single direction, it may still hold that *Growth* is dependent on *Jill*, denoted as $\mathit{Growth} \not\perp\!\!\!\perp \mathit{Jill}$, if we do not condition on *Water*.

After we briefly discussed the causal Markov condition and the faithfulness assumption, we will explain how to learn causal structures under those assumptions. In particular, assuming that both assumptions hold, we are able to 1) distinguish between direct and dependencies for each node, that is, infer the true undirected network, and 2) infer some of the edge directions. We will explain how both can be achieved by summarizing the main ideas of the Peter and Clark (PC) algorithm (Spirtes et al., 2000). To obtain the undirected network, the algorithm starts with a fully connected graph, that is, each pair of nodes is connected through an edge. Based on this graph, we start to delete edges based on the faithfulness assumption. We delete an edge between a pair of nodes X and Y, if X and Y can be rendered independent by conditioning on a set of random variables $\boldsymbol{S}$, which does not include X and Y. In the office plant network, we can remove the edge between *Light* and *Plan* since $\mathit{Light} \perp\!\!\!\perp \mathit{Plan}$. We can further delete the edge between *Jill* and *Growth* since $\mathit{Jill} \perp\!\!\!\perp \mathit{Growth} \mid \mathit{Water}$ and continue until we cannot detect further independencies.

To infer some of the edge directions, it suffices to identify v-structures. A v-structure describes a triple of nodes in which two non-adjacent nodes jointly cause the third node, e.g. $X \rightarrow Y \leftarrow Z$. In particular, if faithfulness holds, X and Z are dependent given Y, even if we additionally condition on any other node in the graph. Vice versa, X and Z can be rendered independent if we do not condition on Y (Spirtes et al., 2000), and hence, we can identify this v-structure. As an example, consider the triple $\mathit{Water} \rightarrow \mathit{Growth} \leftarrow \mathit{Light}$. Clearly, the sunlight is independent of the amount of water provided to the plant. By conditioning on *Growth*, both quantities will become conditionally dependent, which allows us to identify the v-structure and hence infer the corresponding edge directions. We repeat this procedure for each unshielded triple (X and Y are not adjacent) in the undirected graph, which we determined in the previous step. As a result, we identified the correct undirected graph and all v-structures. Subsequently, it might be possible to further infer some edge directions due to the acyclicity of the graph (Meek, 1995a). Such a partially directed graph represents the Markov equivalence class of the true DAG, provided that faithfulness and the causal Markov condition hold.

Although this sounds appealing in theory, the above framework contains quite a list of ifs and buts. First, a lot of the framework heavily depends on the faithfulness assumption to hold. However, as we will show below, there exist causal mechanisms that violate this assumption and thus lead to wrong inferences. Second, the entire construction of the graph relies on correctly detecting (in)dependencies. In practice, small sample sizes, complex generating mechanisms and varying data types can make this task difficult. Last, we can only infer the edge directions up to the Markov equivalence class and need to leave some of the edges undirected. These are precisely the research questions that we study in this thesis, as we elaborate below.

1.2 Research Questions

In the following, we discuss the research questions that we investigate in this dissertation, focusing on the faithfulness assumption, conditional independence testing and cause-effect inference beyond Markov equivalent DAGs.

The faithfulness assumption is a very practical assumption, which is known to hold in simple systems such as linear Gaussian models (Meek, 1995b). However, there exist generating mechanisms such as deterministic relations, cancelling paths or xor structures, which violate this assumption (Ramsey et al., 2006). As a result, algorithms that rely on faithfulness will fail to recover the corresponding edges. In this thesis, we investigate faithfulness violations induced by xor structures, which we explain below.

Let us consider the office plant network again, for which we will now define the generating mechanism in more detail. In particular, the office plan, which is due to a pandemic, should reduce the number of people showing up at the office at the same time, but should also be fair. Hence, the boss of Jack and Jill creates the office plan by throwing an unbiased coin for each employee to determine whether he or she is allowed to go to the office on this day. We also assume that the actual presence of Jack and Jill may occasionally deviate from this plan. Further, Jack and Jill almost always water the plant when they are alone in their office. We model the amount of water they give the plant with a Gaussian distributed random variable. Surprisingly, however, when both Jack and Jill are at the office on the same day, they think that the other person already watered the plant and hence the plant will predominantly not be watered if both are at the office. Consequently, the process of watering the plant follows an exclusive or (xor) structure dependent on the presence of Jack and Jill. This mechanism can be described as follows

Jack	*Jill*	*Plant Watered*
0	0	0
1	0	1
0	1	1
1	1	0

where a “0” denotes that Jack respectively Jill was not at the office and a “1” that they were present. Similarly, “0” denotes that the plant was not watered and “1” that it was. Due to the office plan, the probability that Jack is at the office it roughly one half and the same holds for Jill. As a result, the xor relation will induce an independence between *Jack* and *Plant Watered* as well as *Jill* and *Plant Watered*, as we show in Chapter 2. If we assume that Jack and Jill provide the plant roughly the same amount of water, we will get that *Jack* $\perp\!\!\!\perp$ *Water* as well as *Jill* $\perp\!\!\!\perp$ *Water*. Thus, if we rely on the faithfulness assumption, we would delete the edges between *Jack*, *Jill* and *Water* in the first phase of the PC algorithm. Hence, we would infer the wrong DAG structure, which leads us to the first research question.

Question 1 *How can we discover causal DAGs in the presence of faithfulness violations induced by xor-type relations?*

To answer this question, we thoroughly analyze the dependence structures induced by xor-relations in Chapter 2 and propose a weaker faithfulness assumption that considers triple interactions. In particular, we utilize the fact that xor-relations like the one described above can be detected when observing all three nodes. Subsequently, we define a sound orientation rule, which allows us to infer some of the edge directions within such a structure if it is embedded in a larger graph. To provide some intuition on how to put our ideas into practice, we provide a sound algorithm to discover local causal structures w.r.t. a target node under our assumptions.

Part II of this thesis is dedicated to conditional independence testing. We discussed that independence tests are used to distinguish direct from indirect dependencies and to detect v-structures. Therefore, it is vital to have access to a reliable conditional independence test. If a test is too lenient, i.e., finds dependencies although there are none, the corresponding discovery algorithm would find spurious edges that do not occur in the true DAG. On the other hand, it is important that a test can detect complex dependencies, such as the ones generated by the xor mechanism, since otherwise, we will not recover all true edges. Ideally, an independence test should be applicable to different kinds of data, as it frequently occurs that not all random variables are of the same type. In the office plant network, for example, the presence of Jill at the office can be modelled as a binary variable. The amount of sunlight that shines on the plant, however, has to contain continuous data points. Of course, independence testing is a well studied-topic (Bergsma, 2004), but there do not exist many tests that are applicable to mixed data types. Hence, we formulate the second research question as follows.

Question 2 *How can we detect (conditional) dependencies among mixed-type random variables that can be discrete, continuous or a mixture of both?*

In this thesis, we propose two approaches that build up towards this goal. In Chapter 3, we develop an information-theoretic approach based on algorithmic independence, which is defined via Kolmogorov complexity (Kolmogorov, 1965). Since Kolmogorov complexity itself is not computable, we propose to estimate this quantity for discrete data via an estimator based on the Minimum Description Length (MDL) principle (Rissanen, 1978). In addition, we prove that our estimator is a consistent estimator of conditional mutual information (CMI), which is a non-parametric measure of dependence. Following up on this result, we propose a consistent estimator of CMI for mixed random variables in Chapter 4. To cope with both discrete and continuous data points, we propose to first discretize the continuous data points using adaptive histograms and then estimate CMI from the discretized data.

In Part III, we focus on those edges that cannot be inferred via v-structures, i.e., our goal is to infer DAGs beyond their Markov equivalence class. If we are able to detect all v-structures, the undirected edges that remain are single edges between two nodes. In the office plant network, for example, the edge between *Plan* and *Jack*, as well as the edge between *Plan* and *Jill*, would fall into this category. For both, we cannot exploit a v-structure to obtain the corresponding edge direction. This specific causal discovery task is also called causal inference, bi-variate causal discovery or simply cause-effect inference. In short, the goal is to determine the edge direction between two dependent random variables X and Y, that is, decide between the two possible graph structures $X \to Y$ and $Y \to X$. Note, however, that we implicitly assume that there exists no unobserved variable Z that causes both X and Y and hence induces the dependence between both variables. Under this premise, we formulate our last research question.

Question 3 *How can we distinguish between the two Markov equivalent DAGs $X \to Y$ and $Y \to X$, and do so with guarantees?*

To distinguish between Markov equivalent DAGs, it is necessary to make assumptions about the generating mechanism. We build upon the postulate, which states that the causal mechanism, that is, the conditional distribution of the effect given the cause, is algorithmically independent of the distribution of the cause (Janzing and Schölkopf, 2010). We explain the theoretical foundations of these concepts in Chapter 5. In addition, we analyze under which conditions this algorithmic independence can be approximated via MDL. To put theory into practice, we propose an MDL-based score for univariate numeric data in Chapter 6, which we use to tell cause from effect on observational data. In particular, we assume that the causal mechanism can be expressed by a mixture of local and global regression functions, for which we encode the parameters and the residuals to get an MDL score. In Chapter 7, we simplify the model to non-linear regression functions and focus on identifiability. That is, we specify under which conditions we are guaranteed to infer the correct causal direction using L_0-regularized regression. In the following chapter, Chapter 8, we generalize the idea presented in Chapter 6 and propose an MDL-based estimator for multivariate mixed-type data, which we instantiate using classification and regression trees (CART).

Last, we round up with a conclusion in Chapter 9.

1.3 Contributions of this Thesis

This thesis is a cumulative dissertation based on the research articles listed in Table 1.1. While the main content of those research articles is included verbatim in this dissertation, we made some modifications to keep the thesis coherent. These include changes in the notation, rewriting introductions and conclusions, removing abstracts, and restructuring related work and preliminary sections, which are shared by multiple chapters. Chapter 5 mainly serves as a preliminary chapter for Chapters 6–8. In addition, Chapter 5 contains a short theoretical analysis, which aims to further clarify the theoretical foundations of Chapters 6–8 in hindsight.

Table 1.1: Publications on which this thesis is based.

publication	used in
Alexander Marx, Arthur Gretton, and Joris M. Mooij. *A Weaker Faithfulness Assumption based on Triple Interactions.* In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2021	Chapter 2
Alexander Marx and Jilles Vreeken. *Testing Conditional Independence on Discrete Data using Stochastic Complexity.* In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2019	Chapter 3
Alexander Marx, Lincen Yang, and Matthijs van Leeuwen. *Estimating Conditional Mutual Information for Discrete-Continuous Mixtures using Multi-dimensional Adaptive Histograms.* In Proceedings of the SIAM International Conference on Data Mining (SDM), 2021	Chapter 4
Alexander Marx and Jilles Vreeken. *Telling Cause from Effect using MDL-based Local and Global Regression.* In Proceedings of the IEEE International Conference on Data Mining (ICDM), 2017	Chapter 6
Alexander Marx and Jilles Vreeken. *Telling cause from effect by local and global regression.* In Knowledge and Information System (KAIS), 2019	Chapter 6
Alexander Marx and Jilles Vreeken. *Identifiability of Cause and Effect using Regularized Regression.* In Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2019	Chapter 7
Alexander Marx and Jilles Vreeken. *Causal Inference on Multivariate and Mixed-Type Data.* In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2018	Chapter 8

This thesis is based on the seven research articles listed in Table 1.1, all of which the author was the first author. The contributions for the paper on which we base Chapter 4 were split half/half with Lincen Yang, the officially second author of the corresponding research article.

Part I

The Faithfulness Assumption

In the first part of this thesis, we focus on one of the most fundamental assumptions in causal discovery, faithfulness. Simply put, faithfulness asserts that each independence found in the distribution corresponds to a separation in the true causal graph. In Chapter 2, we will first give a general introduction to graphical models, which are a key tool for modelling causal graphs and explain the most common assumptions on which many causal discovery algorithms rely. Then we focus specifically on the faithfulness assumption, which enables us to make inferences about the causal graph based on the given probability distribution—or a large i.i.d. sample of the true distribution. We analyze several generating mechanisms that could be part of a causal network, such as xor relations, which violate this assumption and hence cannot be detected when assuming that faithfulness holds. Based on the observation that many such faithfulness violations can be detected when looking at triples of nodes instead of pairs, we propose a weaker assumption that allows us to pick up such mechanisms.

Chapter 2

A Weaker Faithfulness Assumption based on Triple Interactions

Two standard assumptions in causal discovery are the causal Markov condition and the faithfulness assumption (Spirtes et al., 2000). While the former assumes that all separations in the true causal graph G imply independencies in P, the faithfulness assumption is its counterpart. That is, all independencies found in P are due to separations in G. Although both assumptions have great merit for causal discovery algorithms, especially the faithfulness assumption has been criticized in the past (Andersen, 2013; Zhang and Spirtes, 2016).

Despite it was proven that faithfulness violations in causally sufficient linear-Gaussian and discrete acyclic systems occur with Lebesgue measure zero (Meek, 1995b), it has also been shown that given a finite sample, empirical faithfulness violations do appear surprisingly often (Uhler et al., 2013). Even on population level, there exist simple generating mechanisms, as shown in Figure 2.1, that violate faithfulness. For instance, two independent random variables X and Z, that can be modelled by fair coins, together cause Y through a noisy xor relation. As a consequence, all three variables are marginally independent. Following the faithfulness assumption, there should be no edges connecting X, Y and Z in the causal graph—however, there are.

Faithfulness violations like the above have been intensively studied in the past (Ramsey et al., 2006; Zhang and Spirtes, 2008; Spirtes and Zhang, 2014)

This chapter is based on Marx, Gretton, and Mooij (2021a).

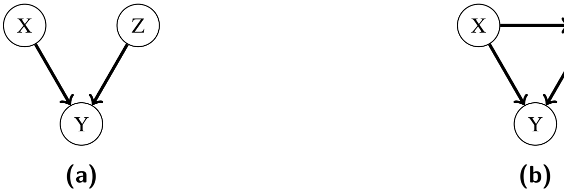


Figure 2.1: Failures of adjacency faithfulness: Assume in graph (a) X, Z are fair independent coins and $Y := (X \oplus Z) \oplus E$, where $\oplus$ is the xor operator and E is a biased coin denoting a noise term. Then X is independent of Y (denoted as $X \perp\!\!\!\perp_P Y$) and $Z \perp\!\!\!\perp_P Y$. Graph (b) could correspond with a linear model where both directed paths from X to Y cancel s.t. $X \perp\!\!\!\perp_P Y$, but $X \not\perp\!\!\!\perp_P Z$ and $Z \not\perp\!\!\!\perp_P Y$.

and several weaker assumptions such as adjacency faithfulness (Spirtes et al., 2000), P-minimality (Pearl, 2009), SGS-minimality (Spirtes et al., 2000) and frugality (Forster et al., 2017), which we review in Section 2.2.3, have been proposed. Although faithfulness violations induced by xor-type relations—i.e., both parents are marginally independent of the child node—can be detected by most of the above approaches, they do not analyze under which conditions the DAG structure can be recovered once such violations have been detected.

To overcome this limitation, we propose a new assumption that we call 2-*adjacency faithfulness*, which allows us to both detect such faithfulness violations and partially infer the underlying DAG structure under certain conditions. We start by explaining the standard concepts and notation in Section 2.1 and review failures of adjacency faithfulness as well as related work in Section 2.2. Then, we study the causal structure of xor-type connections in Section 2.3 and propose 2-adjacency faithfulness in Section 2.4. To partially infer causal DAGs that may contain such generating mechanisms, we introduce a sound orientation rule in Section 2.5. Further, we show under which assumptions on the distribution this rule is applicable—which we formalize as the 2-orientation faithfulness assumption—and analyze its failure cases. As a proof of concept, we introduce a modification of the Grow and Shrink (GS) algorithm (Margaritis and Thrun, 2000) in Section 2.6 and show it correctly identifies the Markov blanket of a target node under strictly weaker assumptions than faithfulness. In addition, we give some intuition on how to extend well-known causal discovery algorithms based on our new assumptions.

2.1 DAGs and Independence

In this section, we define our notation and provide definitions for separations on graphs and independence w.r.t. a probability distribution.

2.1.1 Causal Graphs

A causal *directed acyclic graph* (DAG) G over a set of random variables $\boldsymbol{V}$ with joint distribution P is defined such that each pair of nodes that is adjacent in G is causally related. For simplicity, we will use the random variables $\boldsymbol{V}$ to also refer to the nodes of the graph. A directed edge $X \rightarrow Y$ in G between two nodes representing the random variables $X, Y \in \boldsymbol{V}$ indicates that X is a *direct cause* or *parent* of Y and that Y is a *direct effect* or *child* of X. Accordingly, we denote the set of all parents of $X \in \boldsymbol{V}$ with $\mathrm{Pa}(X)$, the set of all children with $\mathrm{Ch}(X)$ and the set of parents and children with $\mathrm{PC}(X) := \mathrm{Pa}(X) \cup \mathrm{Ch}(X)$. Further, we write $\mathrm{An}(X)$ for the set of *ancestors* and $\mathrm{De}(X)$ for the set of all *descendants* of X, where X is an ancestor and descendant of itself. Respectively, we refer to the *non-descendants* of X as $\mathrm{Nd}(X) := \boldsymbol{V} \backslash \mathrm{De}(X)$. Last, the *Markov blanket* of a variable X is defined as $\mathrm{MB}(X) := \mathrm{PC}(X) \cup \mathrm{Sp}(X)$, where $\mathrm{Sp}(X)$ are the spouses of X, that is, nodes that share a child node with X. Importantly, X is d-separated of any other node in the graph given its Markov blanket and $\mathrm{MB}(X)$ is the smallest such set.

DAGs are used to represent causal graphs under the assumption of acyclicity, no selection bias, and *causal sufficiency*, that is, it is assumed that no two variables $X, Y \in \boldsymbol{V}$ are caused by a confounder Z which is not in the set of observed variables $\boldsymbol{V}$. This is also the setup on which we focus in this thesis—i.e., assuming that all relevant variables are observed, that there are no causal cycles and that there has been no conditioning on selection variables. Further, as a short form to summarize a model as defined above, we write $\mathcal{M} = (G, \boldsymbol{V}, P)$.

2.1.2 Independence and Separation

In the following, we define conditional independence in a probability distribution and d-separation in a graph.

Given three sets of probabilistic random variables $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$, where P is the joint distribution over $\boldsymbol{V}$, we denote that $\boldsymbol{X}$ is *probabilistically independent* of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in P as $\boldsymbol{X} \perp\!\!\!\perp_P \boldsymbol{Y} \mid \boldsymbol{Z}$.

D-separation (Pearl, 2009) is defined in terms of paths. A *path* p between X and Y, denoted $p = \langle X, \ldots, Y \rangle$, is a sequence of distinct nodes $X_1, \ldots, X_n$ such that X_i is adjacent to X_{i+1} for $i = 1, \ldots, n-1$, $X_1 = X$ and $X_n = Y$. Further, we call a node C a *collider* on a path $\langle \ldots, X, C, Y, \ldots \rangle$, where C is adjacent to both X and Y, if two arrowheads point to it, that is $X \rightarrow C \leftarrow Y$.

Definition 2.1 (d-Separation) *A path between two vertices X, Y in a DAG is d-connecting given a set $\boldsymbol{Z}$, if*

1. *every non-collider on the path is not in $\boldsymbol{Z}$, and*
2. *every collider on the path is an ancestor of $\boldsymbol{Z}$.*

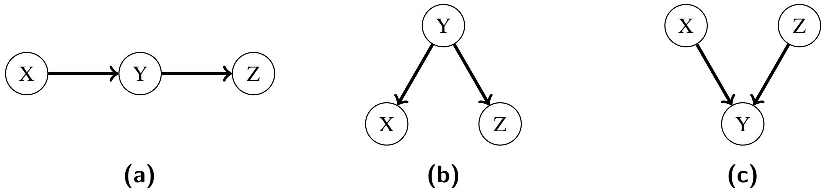


Figure 2.2: The figure shows three possible orientations for the skeleton structure $X - Y - Z$. While the two orientations (a) and (b) are Markov equivalent, since they have the same skeleton structure and the same v-structures (none), (c) is not Markov equivalent to (a) or (b).

If there is no path d-connecting X and Y given $\boldsymbol{Z}$, then X and Y are d-separated given $\boldsymbol{Z}$. Sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated given $\boldsymbol{Z}$, if for every pair X, Y, with $X \in \boldsymbol{X}$ and $Y \in \boldsymbol{Y}$, X and Y are d-separated given $\boldsymbol{Z}$.

As shorthand notation for separations on a DAG G, we write $\boldsymbol{X} \perp\!\!\!\perp_G \boldsymbol{Y} \mid \boldsymbol{Z}$ if $\boldsymbol{X}$ is *d*-separated from $\boldsymbol{Y}$ given $\boldsymbol{Z}$. Building upon this notion, we can now formally define a Markov equivalence class, which is often defined using the notion of *v-structures*, which is just a different way to define a collider structure like $X \rightarrow C \leftarrow Y$, for which X and Y are not adjacent.

Definition 2.2 (Markov Equivalence (Pearl, 2009)) *Two DAGs G_1 and G_2 are said to be Markov equivalent, if and only if they have the same skeleton (underlying undirected graph) and the same v-structures.*

A minimal example to explain Markov equivalence is provided in Figure 2.2. The simple chain graph $X \rightarrow Y \rightarrow Z$ and the common cause graph $X \leftarrow Y \rightarrow Z$ are Markov equivalent, since they do not contain a collider and share the same skeleton. On the contrary, a simple v-structure like $X \rightarrow Y \leftarrow Z$ is not Markov equivalent to the previous two.

Another useful set of tools for inferences on graphs and distributions, which we will need for some of our proofs, are the graphoid axioms (Dawid, 1979; Spohn, 1980; Geiger et al., 1990).

Definition 2.3 (Graphoid Axioms) *Let $\mathcal{M} = (G, \boldsymbol{V}, P)$, with $\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$. The (semi-)graphoid axioms are ($\perp\!\!\!\perp$ denotes $\perp\!\!\!\perp_P$ and $\perp\!\!\!\perp_G$)*

1. *Symmetry: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z} \Rightarrow \boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X} \mid \boldsymbol{Z}$.*
2. *Decomposition: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \mid \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$.*
3. *Weak Union: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \mid \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{W} \cup \boldsymbol{Z}$.*
4. *Contraction: $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{W} \cup \boldsymbol{Z}) \wedge (\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \mid \boldsymbol{Z}) \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \mid \boldsymbol{Z}$.*

For separations only on G, the graphoid axioms include (only for $\perp\!\!\!\perp_G$).

5. *Intersection:* $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{W} \cup \boldsymbol{Z}) \wedge (\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \mid \boldsymbol{Y} \cup \boldsymbol{Z}) \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \mid \boldsymbol{Z}$*, for any pairwise disjoint subsets* $\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$.
6. *Composition:* $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}) \wedge (\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \mid \boldsymbol{Z}) \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \mid \boldsymbol{Z}$.

As an illustration why certain rules only hold for graphs and not generally for probability distributions, consider rule (6) and Figure 2.1 (a) again. From the distribution induced by the xor, we find that $Y \perp\!\!\!\perp_P X$ and $Y \perp\!\!\!\perp_P Z$ but we cannot conclude that $Y \perp\!\!\!\perp_P \{X, Z\}$. If, however, in a graph Y is *d*-separated from X and from Z then Y is *d*-separated from the set $\{X, Z\}$.

We round up this section by defining the causal Markov condition (CMC) and the local Markov condition for DAGs (Spirtes et al., 2000).

Definition 2.4 (Causal Markov Condition) *Let $\mathcal{M} = (G, \boldsymbol{V}, P)$, the causal Markov condition holds, if every d-separation imposed by G implies an independence in P.*

Definition 2.5 (Local Markov Condition) *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$, for each node $X \in \boldsymbol{V}$ it holds that X is d-separated from all non-descendants of X given the parents of X.*

In the remainder of this chapter, we focus on differences between graphs and the distribution. To this end, the causal Markov condition is an essential assumption for causal discovery. On the other hand, assumptions about what properties of the graph can be inferred based on the given distribution have been weakened over time (Ramsey et al., 2006; Zhang and Spirtes, 2008; Forster et al., 2017). Most commonly known is the faithfulness assumption.

2.2 Adjacency Faithfulness and Faithfulness Violations

To lay out the problem, we first explain faithfulness and adjacency faithfulness, then examine when those could fail and give a summary about the most relevant related approaches that use weaker assumptions.

The faithfulness assumption is one of the core assumptions made by most causal discovery algorithms (Spirtes et al., 2000) and it can be seen as the inverse assumption to CMC—i.e., assuming that all independencies found in P imply a *d*-separation in the causal graph. Adjacency faithfulness is a slightly weaker assumption.

Definition 2.6 (Adjacency Faithfulness) *Given the triple $\mathcal{M} = (G, \boldsymbol{V}, P)$, if $X, Y \in \boldsymbol{V}$ are adjacent in G, then they are probabilistically dependent given any subset $\boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, Y\}$.*

Alternatively, we could turn this definition around by stating that if we find a conditional independence in P, then we assume that there is no edge in the corresponding graph. Assuming adjacency faithfulness ensures that we recover the correct skeleton graph (i.e., the undirected graph). Correct detection of the skeleton together with the correct identification of all collider structures ensures that the detected graph is in the Markov equivalence class of the true graph (Verma and Pearl, 1991). The latter is ensured by additionally assuming that orientation faithfulness holds (Zhang and Spirtes, 2008).

Definition 2.7 (Orientation-Faithfulness) *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$. Let the path $\langle X, Y, Z\rangle$ be unshielded in G, that is X is adjacent to Y and Y is adjacent to Z, but X is not adjacent to Z.*

1. *If $X \rightarrow Y \leftarrow Z$, then $X \not\perp\!\!\!\perp_P Z$ given any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains Y; otherwise*
2. *X and Z are dependent conditional on any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that does not contain Y.*

The bottleneck here is the adjacency faithfulness assumption, as many causal discovery algorithms such as PC (Spirtes et al., 2000) or GES (Chickering, 2002) rely on finding adjacent nodes either by checking for marginal dependencies or adding single edges based on adjacency faithfulness and CMC. If one is willing to assume that those assumptions hold, then any violation of orientation faithfulness can be detected as shown by Zhang and Spirtes (2008). However, adjacency faithfulness can be violated in many ways, e.g. by xor-type connections, path cancellations, or deterministic relations. We briefly explain the first two below, as they are relevant for the remainder. For deterministic relations and finite sample failures, see Lemeire et al. (2012).

2.2.1 Xor-Type Relations

In this chapter, we focus on xor-type relations. That is, given a triple of nodes $X, Y, Z \in \boldsymbol{V}$ such that $X \rightarrow Y \leftarrow Z$, where at least one of the causal edges cannot be detected by marginal dependence, but only by looking at the joint distribution over X, Y and Z. The key here is that either parent of Y might not be dependent on Y, but only by considering both parents, we can detect the dependence. To illustrate this, consider the following example where we describe a noisy xor with an unobserved noise variable that is modelled with a biased coin as it is common for binary causal structures (Inazumi et al., 2011).

Example 2.1 *Let $\mathcal{M} = (G, \boldsymbol{V}, P)$ be a causal model. Given variables $X, Y, Z \in \boldsymbol{V}$ such that $X \rightarrow Y \leftarrow Z$ in G and there is no edge connecting X and Z, as in Figure 2.1(a), where X, Z are fair independent coins. Their common effect Y is generated as $Y := (X \oplus Z) \oplus E$, where $\oplus$ denotes xor—i.e., $X \oplus Z := (X + Z)$*

mod 2—*and E is a biased coin with $P(E = 1) = p$, where $0 \leq p < \frac{1}{2}$ and $E \perp\!\!\!\perp_P \{X, Z\}$. Hence, $X \not\perp\!\!\!\perp_G Y$, $Z \not\perp\!\!\!\perp_G Y$, however, due to the xor, we have that $X \perp\!\!\!\perp_P Y$ and $Z \perp\!\!\!\perp_P Y$. Both edges violate adjacency faithfulness. If we were to check the joint distribution, we can find that $Y \not\perp\!\!\!\perp_P \{X, Z\}$, or $X \not\perp\!\!\!\perp_P Z \mid Y$, since we get that $P(X = 1, Z = 1, Y = 1) = \frac{p}{4}$, where $P(X = 1, Z = 1) \cdot P(Y = 1) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$. Those terms being equal would only hold if $p = \frac{1}{2}$, which we excluded by assumption.*

Similar examples, where the marginal dependencies might be hard to detect can be found for continuous data (Sejdinovic et al., 2013), as we show in Figure 2.3. In this mechanism, the effect is modelled through a sign function with exponential noise, where both causes are Gaussian distributed.

2.2.2 Cancelling Paths

A minimal example of cancelling paths was given by Hesslow (1976) and is illustrated with the causal graph shown in Figure 2.1(b). In Hasslow's example taking birth control pills (X) can influence the risk of getting thrombosis (Y) via two paths. It has a direct effect and also taking the pills reduces the chance of pregnancy (Z), which itself is a cause of thrombosis. However, the causal effects induced by those paths cancel such that $X \perp\!\!\!\perp_P Y$ even though $X \not\perp\!\!\!\perp_G Y$. As an example for a mechanism that induces such a cancellation, consider a linear Gaussian system in which $Z := \alpha X$, $Y := \beta Z - \gamma X$ and $\gamma = \alpha\beta$. This failure of faithfulness was shown to be undetectable since X will be dependent on Y given Z and hence the graph $X \rightarrow Z \leftarrow Y$ is also a valid graph for those independencies—i.e., Markov equivalent (Zhang and Spirtes, 2008). There exist cancelling paths that consist of more than three variables, which are detectable, e.g., if Z is not adjacent to Y, but there is a path $Z \rightarrow W \rightarrow Y$ (Zhang and Spirtes, 2008).

2.2.3 Weaker Assumptions

In the following, we discuss different approaches on how to relax faithfulness.

Two well-studied assumptions are P-minimality (Pearl, 2009) and SGS-minimality (Spirtes et al., 2000). While the former states that from all DAGs that satisfy the causal Markov condition w.r.t. P, the DAG that entails most conditional independence statements is preferred. The latter assumes that no proper subgraph of the true DAG fulfils the causal Markov condition w.r.t. to P. From both assumptions, SGS-minimality is the weaker assumption (Zhang, 2013). In a different line of research, it was shown that SGS-minimality suffices for causal discovery approaches based on the assumption that the effect is generated through an additive noise function of the causes (Peters et al., 2014). We discuss additive noise models in more detail in the last part of this thesis.

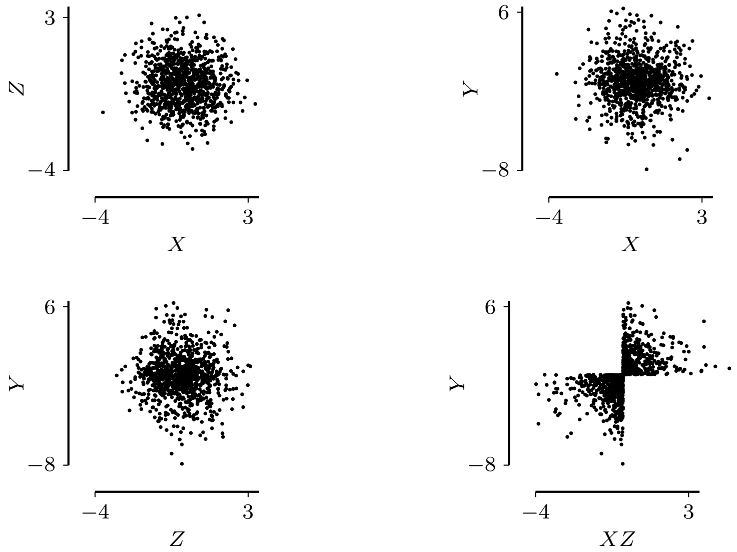


Figure 2.3: Sample data for the collider graph $X \rightarrow Y \leftarrow Z$, where $X, Z \sim N(0,1)$ are i.i.d. and $Y := \text{sign}(XZ) \cdot E$, with $E \sim \text{Exp}(\frac{1}{\sqrt{2}})$. The dependence is only detectable by considering X, Y and Z jointly.

A more recent approach by Forster et al. (2017) introduces the concept of *frugality*, which is a stronger assumption than both minimality assumptions. The authors define a DAG G to be more frugal than G', if G contains fewer edges than G'. A maximally frugal DAG uses only as many edges as are necessary to satisfy the causal Markov condition. To determine maximally frugal graphs, one has to consider all causal orderings of the variables, which is rather costly, but can be solved using permutation algorithms (Raskutti and Uhler, 2018). Another approach to discover causal graphs based on frugality, or any of the above assumptions is based on boolean satisfiability (SAT) solvers (Zhalama et al., 2017). Here, we introduce 2-adjacency faithfulness, which allows us to find xor-type relations, some faithfulness violations induced by cancelling paths and all relations that are detectable by assuming adjacency faithfulness. We conjecture that 2-adjacency faithfulness is a slightly stronger assumption than frugality since frugality considers all permutations and not only triples (Forster et al., 2017). However, this can also be an advantage, since we only need to check all triples to detect 2-associations, instead of all permutations. In addition, we extend existing work by providing a sound orientation rule that can be used to infer the edges within a 2-association, if they appear in a larger graph.

In the next section, we discuss xor-type relations in more detail. We use those structures as an example to illustrate one of the main properties of 2-associations, that we describe in Theorem 2.1.

2.3 Unfaithful Triples

We first define what we call an unfaithful triple[1] and its properties, and then argue why such a triple a) violates adjacency faithfulness and b) even if detected, the underlying DAG structure cannot be uniquely determined without further information.

Definition 2.8 (Unfaithful Triple) *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$ and three distinct nodes $X, Y, Z \in \boldsymbol{V}$: if X, Y and Z are marginally independent but not mutually independent in P, we call $\{X, Y, Z\}$ an unfaithful triple w.r.t. P.*[2] *If further for each distinct pair of nodes $A, B \in \{X, Y, Z\}$:*

$$\forall \boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, Y, Z\} : A \not\perp\!\!\!\perp_P B \mid \boldsymbol{S} \cup \{X, Y, Z\} \backslash \{A, B\}\,,$$

we call $\{X, Y, Z\}$ a minimal unfaithful triple.

The first example for such a triple for three binary random variables was given by Bernstein (1927), which is equivalent to our noisy xor example. The minimality condition ensures that the three nodes are connected by a path of length two, as we will show below. We start by showing that if three random variables $\{X, Y, Z\}$ are marginally independent, finding a dependence between all three variables, e.g. $X \not\perp\!\!\!\perp_P \{Y, Z\}$, implies that also $Y \not\perp\!\!\!\perp_P \{X, Z\}$ and $Z \not\perp\!\!\!\perp_P \{X, Y\}$.

Lemma 2.1 *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$, let $\{X, Y, Z\} \subseteq \boldsymbol{V}$ form an unfaithful triple in P, then $X \not\perp\!\!\!\perp_P \{Y, Z\}$, $Y \not\perp\!\!\!\perp_P \{X, Z\}$ and $Z \not\perp\!\!\!\perp_P \{X, Y\}$, which in addition implies that $X \not\perp\!\!\!\perp_P Y \mid Z$, $X \not\perp\!\!\!\perp_P Z \mid Y$ and $Y \not\perp\!\!\!\perp_P Z \mid X$.*

Proof: *Assume that w.l.o.g. $X \not\perp\!\!\!\perp_P \{Y, Z\}$ is violated. By weak union, we get $X \perp\!\!\!\perp_P Y \mid Z$ which is equivalent to $Y \perp\!\!\!\perp_P X \mid Z$, using symmetry. We know that $Y \perp\!\!\!\perp_P Z$. By contraction, we get that $Y \perp\!\!\!\perp_P \{X, Z\}$. Similarly, we conclude that $Z \perp\!\!\!\perp_P \{X, Y\}$. Altogether, this implies that X, Y, Z would be independent, which is a contradiction. Next, we can see that each pair of joint dependence and marginal independence, e.g. $X \not\perp\!\!\!\perp_P \{Y, Z\}$ and $X \perp\!\!\!\perp_P Z$, implies a conditional dependence, e.g. $X \not\perp\!\!\!\perp_P Y \mid Z$, by contraction.* □

[1] Ramsey et al. (2006) used the term unfaithful triple for the non-detectable faithfulness violation explained in Section 2.2.2.

[2] Not mutually independent implies that $X \not\perp\!\!\!\perp_P \{Y, Z\}$, $Y \not\perp\!\!\!\perp_P \{X, Z\}$ or $Z \not\perp\!\!\!\perp_P \{X, Y\}$.

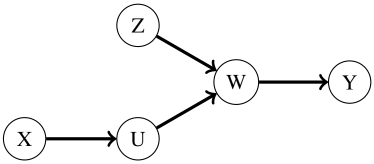


Figure 2.4: Assume that $\{X, Y, Z\}$ form an unfaithful triple. Since X is d-separated from Y given U and Z, they do not form a minimal unfaithful triple. Neither do $\{U, Y, Z\}$, since Y can be d-separated from U given $\{W, Z\}$. Thus, only $\{U, W, Z\}$ can be a minimal unfaithful triple.

Consider Example 2.1. Since X, Y, Z form an unfaithful triple, we can infer from Lemma 2.1 that each pair is conditionally dependent given the third node. As there are no other nodes in the graph, X, Y, Z must form a minimal unfaithful triple. Next, we show that (minimal) unfaithful triples must be connected in the causal graph.

Lemma 2.2 *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$, let $\{X, Y, Z\} \subseteq \boldsymbol{V}$ form an unfaithful triple in P. If CMC holds, each node in the triple is d-connected to at least one other node in the triple by a path in G.*

PROOF: *Assume w.l.o.g. that X is d-separated from Y and Z in G—i.e., $X \perp\!\!\!\perp_G Y$ and $X \perp\!\!\!\perp_G Z$. By applying the composition axiom, we get $X \perp\!\!\!\perp_G \{Y, Z\}$. If we apply the causal Markov condition, we get that $X \perp\!\!\!\perp_P \{Y, Z\}$, which is a contradiction to our assumption.* □

Further, we show that a minimal unfaithful triple has to contain a collider on a path of length two that connects all three nodes in the triple, e.g. $X \to Y \leftarrow Z$ (see Figure 2.4). To do that, we first show a more general statement.

Theorem 2.1 *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$ with three distinct nodes $X, Y, Z \in \boldsymbol{V}$ and assume that CMC holds. If $\forall \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X, Y, Z\}$ it holds that $X \not\perp\!\!\!\perp_P Y \mid Z \cup \boldsymbol{S}$, $X \not\perp\!\!\!\perp_P Z \mid Y \cup \boldsymbol{S}$ and $Y \not\perp\!\!\!\perp_P Z \mid X \cup \boldsymbol{S}$, then one of the three nodes is a collider on a path of length two between the two other nodes, e.g. $X \to Y \leftarrow Z$ in G.*

PROOF: *There must be (at least) one node in $\{X, Y, Z\}$ that is not an ancestor of any of the other nodes, say $Z \notin An(X)$ and $Z \notin An(Y)$, because of acyclicity. In other words, $X \notin De(Z)$ and $Y \notin De(Z)$. The local Markov property states that $Z \perp\!\!\!\perp_G Nd(Z) \mid Pa(Z)$ and hence in particular $Z \perp\!\!\!\perp_G \{X, Y\} \mid Pa(Z)$. Further, if $|Pa(Z) \cap \{X, Y\}| < 2$, we get a contradiction with the assumed conditional dependences. Hence $\{X, Y\} \subseteq Pa(Z)$ and $X \to Z \leftarrow Y$ is in G.* □

The theorem only states that there exists a collider, e.g. $X \rightarrow Y \leftarrow Z$, but not whether this path is shielded or not. Since we do not assume any marginal dependence or independence in Theorem 2.1, we can derive that the same statement holds for a minimal unfaithful triple. Notice that for a minimal unfaithful triple each pair of nodes is marginally independent, which implies that there is no way to decide which of the three possible collider structures corresponds with the causal graph in the absence of further information.

Knowing that a minimal unfaithful triple has to contain a collider in G, it is obvious that such a structure violates adjacency faithfulness, as none of the edges is represented by a marginal dependence in P. The key point is that we can detect such interactions by taking multiple parents into account. In the following, we define a weaker assumption that allows us to detect and infer causal graphs that contain such faithfulness violations.

2.4 2-Adjacency Faithfulness

To define our new assumption, we first need to define associations between a single node and a set of nodes.

Definition 2.9 (k-Association) *Let P be the joint distribution of a set of random variables $\boldsymbol{V}$.*

1. *Given distinct $X, Y \in \boldsymbol{V}$, we say that X is 1-associated to Y, if for each subset $\boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, Y\}$ it holds that $X \not\perp\!\!\!\perp_P Y \mid \boldsymbol{S}$.*
2. *Given distinct $X, Y_1, Y_2 \in \boldsymbol{V}$, X is 2-associated to $\{Y_1, Y_2\}$ if for each subset $\boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, Y_1, Y_2\}$ it holds that*
   i) $X \not\perp\!\!\!\perp_P Y_1 \mid \boldsymbol{S} \cup Y_2$,
   ii) $X \not\perp\!\!\!\perp_P Y_2 \mid \boldsymbol{S} \cup Y_1$ and
   iii) $Y_1 \not\perp\!\!\!\perp_P Y_2 \mid \boldsymbol{S} \cup X$.

We call X strictly 2-associated to the set $\{Y_1, Y_2\}$, if X is 2-associated to $\{Y_1, Y_2\}$ and X is not 1-associated to either Y_1 or Y_2.

In other words, k-associations relate to two types of dependencies: certain conditional dependencies between pairs of variables (1-associations) and between triples (2-associations). For readability, we use a shorthand notation and write $X -_2 \{Y, Z\}$ if X is 2-associated to Y and Z resp. $X -_1 Y$ if X is 1-associated to Y. We denote a strict 2-association by "$\overset{s}{-}_2$". If we refer to a set $\boldsymbol{Y}$ that contains at most two elements and we want to express that X is either 1- or 2-associated to this set, we write $X -_{\leq 2} \boldsymbol{Y}$. Similarly, we write $X \overset{s}{-}_{\leq 2} \boldsymbol{Y}$, if X is 1- or strictly 2-associated to $\boldsymbol{Y}$.

Pairwise dependencies can occur for example in a simple chain $X \rightarrow Y \rightarrow Z$, where no adjacency failure occurs. In this case, $X -_1 Y$ and $Y -_1 Z$. Triple interactions that match the definition of 2-associations, however, need to have a specific structure. As we saw in Theorem 2.1, 2-associations always contain a collider. Thus, a chain graph or a common cause structure does not induce a 2-association. On the other hand, the minimum unfaithful triple in Example 2.1 matches the definition, since $X \overset{s}{-}_2 \{Y, Z\}$, $Y \overset{s}{-}_2 \{X, Z\}$ and $Z \overset{s}{-}_2 \{X, Y\}$. In general, strict 2-associations describe collider structures such as $X \rightarrow Y \leftarrow Z$ for which at least one of the edges violates adjacency faithfulness. If faithfulness holds, a collider structure induces a 2-association, but not a strict 2-association. We use this intuition for our new assumption.

Definition 2.10 (2-Adjacency Faithfulness) *Given $\mathcal{M} = (G, \boldsymbol{V}, P)$, for all $X, Y \in \boldsymbol{V}$, where X and Y are adjacent in the generating DAG G, there exists $\boldsymbol{Y} \subseteq MB(X)$, with $Y \in \boldsymbol{Y}$, s.t. $X \overset{s}{-}_{\leq 2} \boldsymbol{Y}$.*

The main idea here is to weaken adjacency faithfulness such that if a marginal dependence is not present, i.e., adjacency faithfulness is violated, there will be a dependence in combination with a parent, child or spouse. If adjacency faithfulness is not violated, we will not find any strict 2-associations and our assumption reduces to adjacency faithfulness. By also considering strict 2-associations, however, we can discover a larger spectrum of causal mechanisms.

The textbook example for a mechanism that violates faithfulness but is detectable by assuming 2-adjacency faithfulness is the xor-connection described in Example 2.1. Here, $Y \overset{s}{-}_2 \{X, Z\}$, two parents, while $X \overset{s}{-}_2 \{Y, Z\}$—i.e., a child and a spouse. We could even slightly adapt the mechanism and only model Z using an unbiased coin but use a biased coin for X. In this case, only X is marginally independent of Y, while Z becomes dependent on Y. Moreover, assuming 2-adjacency faithfulness could even allow us to detect some faithfulness violations that are due to cancelling paths. In particular, consider the two paths $X \rightarrow Y$ and $X \rightarrow Z \rightarrow W \rightarrow Y$ that cancel such that $X \perp\!\!\!\perp_P Y$. Since $X \perp\!\!\!\perp_P Y$, $X \perp\!\!\!\perp_P W \mid Z$, X could be strictly 2-associated to the set $\{W, Y\}$. Since we know that a 2-association contains a collider and we can neither find a 1-association to Y or W, we know that there has to be an edge violating adjacency faithfulness.

It is not possible to rely on orientation faithfulness when dealing with strict 2-associations. Although we know that a strict 2-association has to contain a collider, we do not know the skeleton structure within the triple and hence cannot apply orientation faithfulness. Next, we show that we can sometimes identify the collider if such a triple occurs in a larger graph.

2.5 2-Orientation Faithfulness

So far, we showed how we can detect unfaithful triples from conditional (in)dependence statements under the weaker assumption of 2-adjacency faithfulness.

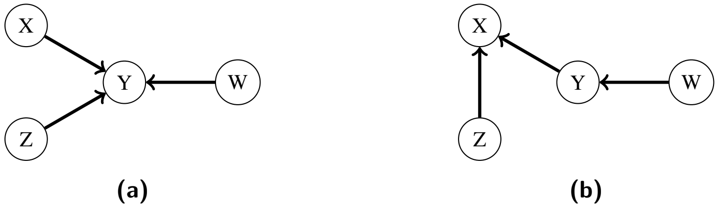


Figure 2.5: In both distributions $Y \overset{s}{-}_2 \{X, Z\}$ and $Y -_1 W$. In the graph shown in (a) Y is a collider on all paths between $\{X, Z\}$ and W, whereas in (b) Y is a non-collider.

Now imagine that we want to use this knowledge for causal discovery. If we observe an isolated triple that follows the dependence structure of the noisy xor, we can only tell that there is a collider. However, if we are given more information, we are able to break this symmetry.

Example 2.2 *Consider that X and Z are unbiased coins as in the noisy xor example. In addition, there is a binary variable W with $P(W = 1) = p$, where $0 < p < 1$ and an unobserved binary noise variable E with $P(E = 1) = q$, where $0 < q < \frac{1}{2}$. Now we generate Y as*

$$Y := ((X \oplus Z) \wedge W) \oplus E\ ,$$

where E, W, X and Z are drawn independently. The requirements for p ensure that W is dependent on Y and the requirements on E ensure that the dependencies are non-deterministic ($q \neq 0$) and evident without observing E ($q \neq \frac{1}{2}$). The corresponding causal graph is given in Figure 2.5(a). From the induced dependencies, that we derive in detail in Appendix A.1, we can now obtain an asymmetry. In particular, $\{X, Y, Z\}$ form a minimal unfaithful triple, but only Y is dependent on W, whereas $\{X, Z\} \perp\!\!\!\perp_P W$ and due to the xor, $X \perp\!\!\!\perp_P W \mid Y$ as well as $Z \perp\!\!\!\perp_P W \mid Y$. Thus, we can detect that there is no edge between X and W or Z and W since none of these pairs can be 2-associated. However, we do find that $X \not\perp\!\!\!\perp_P W \mid \{Y, Z\}$ and $Z \not\perp\!\!\!\perp_P W \mid \{Y, X\}$. As we will show in Theorem 2.2, we can use this information to identify Y as the collider in the triple and $W \to Y$.

To detect such an asymmetry, it is necessary that the collider in the triple is the effect of another node or pair of nodes. If, for example, X would be the collider in the triple and $W \to Y$ (see Figure 2.5(b)), we cannot find such an asymmetry. To generate that graph we could model Y as a noisy copy of W and construct X with a noisy xor from Y and Z. We still know that W is adjacent to Y, but we cannot direct any of the edges as for example we would find that $X \not\perp\!\!\!\perp_P W \mid Z$, which we would also observe if Z would be the collider in the triple, or if we would flip the edge direction between Y and W—i.e., W would be a noisy copy of Y.

Based on this intuition, we propose an orientation rule that may include causal structures that induce strict 2-associations. To do so, we use a shorthand notation—i.e., write $\boldsymbol{Y} \rightarrow X$, if for each element $Y \in \boldsymbol{Y}$ it holds that $Y \rightarrow X$ and vice versa write $X \rightarrow \boldsymbol{Y}$ if X is a parent of each node $Y \in \boldsymbol{Y}$, that is, $\forall Y \in \boldsymbol{Y} : X \rightarrow Y$.

Definition 2.11 (Orientation Rule) *Let $M := (G, \boldsymbol{V}, P)$ and we are given two disjoint sets $\boldsymbol{X}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ and $Y \in \boldsymbol{V}$, where $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ and $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$, and no $X \in \boldsymbol{X}$ is adjacent to some $Z \in \boldsymbol{Z}$ in G.*

i) If for each pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$ it holds that X is dependent on Z given any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $Y \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$, then $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$,

ii) otherwise, if for each pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$ it holds that X is dependent on Z conditional on any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $(\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$ but does not contain Y, Y is a non-collider on at least one path $\langle X, Y, Z \rangle$ where $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$.

Simply put, the above orientation rule relies on the fact that a (strict) 2-association contains a collider. Either Y is the collider on each path $\langle X, Y, Z \rangle$ between any variable $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$ or Y is one of the parents in at least one of the triples and hence blocks at least one such path. If both sets $\boldsymbol{X}$ and $\boldsymbol{Z}$ only contain a single element, rule i) refers to a "normal" collider, e.g. $X \rightarrow Y \leftarrow Z$ and rule ii) refers either to a chain like $X \rightarrow Y \rightarrow Z$ or to a common cause $X \leftarrow Y \rightarrow Z$. Let us consider Example 2.2 again, where we generated Y as a non-deterministic function of X, Z and W. First, we find that $Y \overset{s}{-}_2 \{X, Z\}$, $Y -_1 W$ and W is not adjacent to X or Z (since W is not 1- or strictly 2-associated to X or Z), which is required to apply our rule. Further, we can apply rule i) since W is dependent on X given any set that includes $\{Y, Z\}$ and W is dependent on Z given any set that includes $\{Y, X\}$. Hence, we can infer that $\{X, Z\} \rightarrow Y \leftarrow W$.

In the following we first show that our orientation rule is sound—i.e., if rule i) or ii) can be applied, we are sure we found the true graph structure—and then analyze the inverse, that is, what assumptions need to hold such that the given graph structure implies the suggested dependence model.

Theorem 2.2 *Assuming that the CMC holds, the orientation rule in Definition 2.11 is sound.*

We provide the proof for Theorem 2.2 in Appendix A.3. We show both rules by contraposition, that is, to show the implication in rule ii) holds, we prove that if the true structure is $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$ (exactly the structure not implied by rule ii)), we can always find a pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$ such that X becomes independent of Z if we condition on a set that includes $(\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$, but does not contain Y. Rule i) can be proven accordingly.

The question that remains is: Does the inverse always hold? For example, if the true graph contains a non-collider structure such as $X \rightarrow Y \rightarrow Z$, will we always find that $X \not\perp\!\!\!\perp_P Z$? The short answer is no. Already when we only assume adjacency faithfulness, it can happen that $X \perp\!\!\!\perp_P Z$ although the true graph is $X \rightarrow Y \rightarrow Z$ and it holds that $X \not\perp\!\!\!\perp_P Y$ and $Y \not\perp\!\!\!\perp_P Z$, which is called failure of transitivity. More generally, assuming that orientation faithfulness holds, such failures will not occur. In the following, we extend this assumption to our setting.

Definition 2.12 (2-Orientation Faithfulness) *Let $M := (G, \boldsymbol{V}, P)$ and we are given disjoint $\boldsymbol{X}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ and $Y \in \boldsymbol{V}$, where $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ and $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$, and no $X \in \boldsymbol{X}$ is adjacent to some $Z \in \boldsymbol{Z}$ in G.*

i) If $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$ is in G, then for each pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$, $X \not\perp\!\!\!\perp_P Z$ given any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $Y \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$,

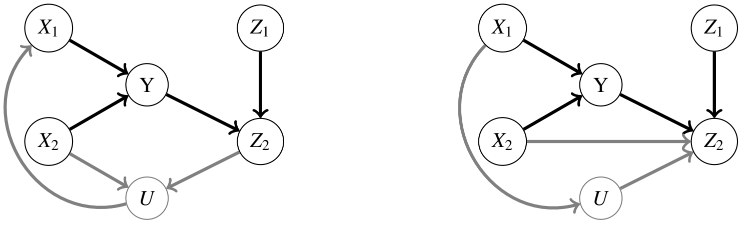
ii) otherwise, for each pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$, $X \not\perp\!\!\!\perp_P Z$ conditional on any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $(\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$, but not Y.

Equivalently to 2-adjacency faithfulness, 2-orientation faithfulness reduces to orientation faithfulness, if both sets $\boldsymbol{X}$ and $\boldsymbol{Z}$ only contain a single element. For orientation faithfulness, it has been shown that all failures can be detected under the assumption that adjacency faithfulness holds (Zhang and Spirtes, 2008). Sadly, an equally strong statement cannot be made for 2-adjacency faithfulness and 2-orientation faithfulness, as we discuss below.

2.5.1 Failures of 2-Orientation Faithfulness

Without any assumptions, we can detect triples for which $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ and $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$, and know by assuming CMC that all 2-associations contain a collider. If further, all paths $\langle X, Y, Z \rangle$ with $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ are unshielded, we can detect if any of the conditions in 2-orientation faithfulness fails. In particular, due to the soundness of our orientation rule, we would detect that none of the conditions in the orientation rule is satisfied if condition i) or ii) in 2-orientation faithfulness fails, as we show in Corollary 2.1.

Yet, we cannot detect all failures of 2-orientation faithfulness. That is due to the fact that we might not always be able to detect whether all paths $\langle X, Y, Z \rangle$ are unshielded. If there is a direct edge between X and Z, we will always find that those nodes are either 1-associated or there exists a third node U such that they are strictly 2-associated (if 2-adjacency faithfulness holds). However, if we find a strict 2-association between X and $\{Z, U\}$ there is no



(a) **(b)**

Figure 2.6: In both figures, $Y \overset{s}{-}_2 \boldsymbol{X} = \{X_1, X_2\}$, $Y \overset{s}{-}_2 \boldsymbol{Z} = \{Z_1, Z_2\}$ (related nodes and edges are marked in black) and $X_2 \overset{s}{-}_2 \{U, Z_2\}$. If we are only given this information, we cannot determine whether the path $\langle X_2, Y, Z_2 \rangle$ is unshielded (a) or shielded (b). While in graph (a), we could safely apply our orientation rule, the shielded graph (b) can be problematic. Due to the directed path from X_1 over U to Z_2 and the adjacency between X_2 and Z_2, each pair $X, Z \in \boldsymbol{X} \times \boldsymbol{Z}$ is now d-connected given $\{Y\} \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$. Thus, the condition for rule i) could hold, although $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$ is not in G.

guarantee that the path is shielded. In particular, if U is the collider between X and Z, the triple is unshielded; but if Z is the collider between X and U, the triple is shielded (see Figure 2.6, in which X refers to X_2 and Z to Z_2). In a causal discovery algorithm, we could try to iteratively infer the DAG structure within such triples until we cannot apply the rule anymore. If we are lucky, we can first infer that $X \rightarrow U \leftarrow Z$ and after that also apply our rule for $\{\boldsymbol{X}, Y, \boldsymbol{Z}\}$. Keeping this exception in mind, we can derive the following corollary from Theorem 2.2.

Corollary 2.1 *Given $M := (G, \boldsymbol{V}, P)$ with $Y \in \boldsymbol{V}$ and $\boldsymbol{X}, \boldsymbol{Z} \subseteq \boldsymbol{V}$, where $\boldsymbol{X} \cap \boldsymbol{Z} = \emptyset$, $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$, $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$ and no pair of nodes $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ is adjacent. Assuming that CMC holds, we can detect if condition i) or ii) of 2-orientation faithfulness fails on the triple $\{\boldsymbol{X}, Y, \boldsymbol{Z}\}$.*

The proof is provided in Appendix A.3. In general, 2-orientation faithfulness might be useful not only for constraint-based causal discovery methods, but also for algorithms that aim to discover the Markov blanket of a target node or permutation-based causal discovery algorithms such as the Sparsest Permutation (SP) algorithm proposed by Raskutti and Uhler (2018). In Appendix A.2, we provide a short discussion from which we conjecture that the SP algorithm can identify the collider pattern even for strict 2-associations like in Figure 2.5(a), if 2-orientation faithfulness holds.

In the next section, we demonstrate how to put theory into practice and propose an algorithm to find the Markov blanket of a target node under 2-adjacency faithfulness.

2.6 A Modified GS Algorithm

As a proof of concept, we propose a simple modification of the Grow and Shrink (GS) algorithm (Margaritis and Thrun, 2000) to discover Markov blankets that can contain strict 2-associations. After that, we briefly discuss further challenges that need to be solved to propose a causal discovery algorithm based on our new assumptions.

The GS algorithm is a simple and theoretically sound causal discovery algorithm, that as a first step identifies the Markov blanket for each node (Margaritis and Thrun, 2000). This step of the algorithm consists of a grow phase, in which we iteratively discover a superset of the Markov blanket of a target node T, and a shrink phase, in which superfluous nodes are pruned.

To make sure that we can detect Markov blankets that contain strict 2-associations, we assume that 2-adjacency faithfulness holds and that we can detect all spouses. For the latter, we assume that a slight variation of the 2-orientation faithfulness assumption holds. In essence, we need to assume that condition i) in 2-orientation faithfulness also holds for shielded triples, which boils down to assuming that the spouses of the target do not cancel each other out, as we explain below.

Assumption 2.1 *Let $M := (G, \boldsymbol{V}, P)$ and we are given two disjoint sets $\boldsymbol{X}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ and $Y \in \boldsymbol{V}$, where $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ and $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$. If $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$ in G, then for each pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$, X is dependent on Z given any subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $Y \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$.*

The above assumption is a relatively lightweight adaption of condition i) in 2-orientation faithfulness. In particular, let $\boldsymbol{X} = \{X, T\}$, where T is the target node. Then all nodes in $\mathbf{Z}$ are spouses of T and even become part of PC(T) if all paths $\langle T, Y, Z \rangle$ for $Z \in \mathbf{Z}$ are shielded. Thus, we would already add those nodes when looking for the parents and children of T. The only complication that may arise is if the second node $X \in \boldsymbol{X}$ is adjacent to a node in $Z \in \mathbf{Z}$ and this adjacency would lead to a cancellation such that Z is only dependent on T if we do not condition on X. The corresponding causal graph consists of the paths $T \rightarrow Y \leftarrow Z$ and $Y \leftarrow X \rightarrow Z$. Since X cannot block the path $\langle T, Y, Z \rangle$, such a scenario seems to be only possible if the causal mechanism that generates Z from X is deterministic. Based on this assumption, we can introduce our adapted GS algorithm.

The generalized GS algorithm is shown in Algorithm 2.1, where we only modified the grow phase to also consider pairs of random variables. This allows us to find nodes to which the target node is strictly 2-associated or spouses to which a child node of T is strictly 2-associated using Assumption 2.1. The shrink phase is not modified and checks if singletons can be removed. It is important to note that we will not remove single nodes of a true strict 2-association

Algorithm 2.1: Modified GS for Markov Blankets

```
input  : Random variables $\boldsymbol{V}$ with joint distribution $P$, Target
         $T \in \boldsymbol{V}$
output: MB($T$)
$\boldsymbol{V}' \leftarrow \boldsymbol{V} \backslash \{T\}$;
$\boldsymbol{S} \leftarrow \emptyset$;
// Grow Phase
while $(\exists X \in \boldsymbol{V}' : T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S}) \vee$
$(\exists X, Z \in \boldsymbol{V}' : T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S} \cup \{Z\})$ do
    $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{X\}$ ;
// Shrink Phase
while $\exists X \in \boldsymbol{S} : T \perp\!\!\!\perp_P X \mid \boldsymbol{S} \backslash X$ do
    $\boldsymbol{S} \leftarrow \boldsymbol{S} \backslash X$ ;
return $\boldsymbol{S}$
```

to T or a child of T, because we do not check for marginal dependencies. For example, assume that $T \overset{s}{-}_2 \{X, Z\}$ and both nodes were added in the grow phase, where X is a child of T and Z the corresponding spouse. If we try to remove X in the shrink phase, we have that $T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S} \backslash X$, since $Z \in \boldsymbol{S}$. Hence, X remains in $\boldsymbol{S}$, as well as Z.

In the following, we show that our proposed algorithm correctly identifies the Markov blanket of a target node assuming that 2-adjacency faithfulness, the causal Markov condition and Assumption 2.1 hold.

Theorem 2.3 *Let $M = (G, \boldsymbol{V}, P)$, and assume 2-adjacency faithfulness, Assumption 2.1 and CMC hold. Algorithm 2.1 identifies MB(T) for $T \in \boldsymbol{V}$.*

We provide the proof in Appendix A.3. For discovering the Markov blanket, we do not need to know the collider of a strict 2-association since it only returns a set of nodes. The more challenging task is to implement our framework to discover causal networks. As an example, the next step in the GS algorithm would be to distinguish the spouses from the parents and children of a node. However, this is not straightforward for strict 2-associations, since we first need to identify the collider in the triple. Similarly, we could extend well-known algorithms such as the PC algorithm (Spirtes et al., 2000) or the GES algorithm (Chickering, 2002) by modifying the skeleton phase, respectively the forward phase such that we can find triple interactions as we did for GS. The edge orientation could be done by first applying the orientation rule in Definition 2.11 and then applying a similar set of rules like Meek's orientation rules (Meek, 1995a).

Alternatively, it was shown that SAT-based discovery algorithms can be easily adapted to weaker assumptions than faithfulness (Zhalama et al., 2017), which is an interesting avenue for future work.

2.7 Conclusion

In this chapter, we showed that we can relax adjacency faithfulness such that we are able to detect edges which would be missed by assuming faithfulness or adjacency faithfulness. In particular, we proposed 2-adjacency faithfulness, which only assumes a dependence between a node X and a set of at most two nodes that are part of the Markov blanket of X. We provided an in-depth analysis of such dependencies and proposed a sound orientation rule, which can infer part of the correct causal structure by detecting collider structures. We complemented this rule with 2-orientation faithfulness, which assumes that if a causal graph contains such collider structures, we will find that the corresponding conditional dependence statements hold in P. As a proof of concept, we showed that we can extend the GS algorithm to find Markov blankets under strictly weaker assumptions than faithfulness.

Although we did not provide an empirical evaluation of our work, we can at least validate that xor-type dependencies like the one shown in Figure 2.3 can be detected by several independence tests, as we show in Chapter 4. Amongst these tests is also the one we propose in the corresponding chapter. However, the question is, how reliable can we find such dependencies or complex dependencies in general, and how high are the chances that we find false positives? In the following part, we attempt to answer these questions and evaluate conditional independence testing via mutual information estimation on both discrete data and mixed-type data.

Part II

Conditional Independence Testing

In this second part, we focus on the other key aspect in constraint-based causal discovery, which is independence testing. The ideal test can pick up complex dependencies, is data-efficient, that is, also detect dependencies on small sample sizes and restrictive enough to not allow for false positives. A natural choice to measure complex dependencies is via (conditional) mutual information (CMI), which is non-parametric and in theory, can pick up any kind of dependence. For discrete data, however, the empirical estimator for CMI overestimates dependencies. To adjust for this bias, we propose to compute the involved entropies via the normalized maximum likelihood instead of using the plug-in maximum likelihood estimator, as described in Chapter 3. Only considering discrete data might be theoretically appealing, but in practice, data can be discrete, continuous or even consist of mixture variables that are partially discrete and partially continuous. To also detect dependencies on such mixed data, we propose to first learn an adaptive histogram model by discretizing the continuous part of each random variable and then estimate CMI from the discretized data, as we show in Chapter 4. Through empirical evaluation, we show that both these estimators perform well in practice and can pick up complex dependencies such as xor-type relations, which we discuss the Part I.

Chapter 3

Independence Testing on Discrete Data

As we showcase in the first part, testing for conditional independence is a crucial part of constraint-based causal discovery algorithms. For instance, in a simple Markov chain $X \rightarrow Z \rightarrow Y$, X and Y may be dependent but are rendered independent given Z. Vice versa, a collider structure such as $X \rightarrow Z \leftarrow Y$ may introduce a dependence between two marginally independent variables X and Y when conditioned on Z. A theoretically appealing way to measure dependencies is through the non-parametric mutual information (MI), since it has several important properties, such as the chain rule, the data processing inequality, and—last but not least—it is zero if (and only if) two random variables are independent of each other (Cover and Thomas, 2012). To also measure conditional dependencies, as needed to detect collider structures, we need to consider conditional mutual information (CMI).

On discrete data, however, the empirical estimator for conditional mutual information is known to overestimate dependencies (Mandros et al., 2017; Paninski, 2003) on small sample sizes. Consequently, if we only have an empirical sample and do not know the true distribution, it is likely that the plug-in estimator is not zero even if the data is generated independently. In extreme cases, the empirical conditional entropy $\hat{H}(X \mid Y)$ can even become zero if the sample space of Y is larger than the number of samples (Mandros et al., 2017). Thus, we could find a functional dependence between X and Y, although they are independent. The same effect translates to CMI.

This chapter is based on Marx and Vreeken (2019a).

To counter this bias of the empirical estimator, several suggestions have been made in the past. Zhang et al. (2010) set a fixed threshold λ, which they use to control for finite sample errors. However, this approach only works when the sample size is known beforehand, as the bias of CMI decreases for larger sample sizes. Most relevant for us are the approaches by Goebel et al. (2005) and Suzuki (2016), which compute a threshold that is aware of both the sample size and the domains of the involved variables. While Goebel et al. (2005) propose to set a threshold based on the gamma distribution, where the degrees of freedom depend on the domain sizes of the variables, Suzuki (2016) chooses a threshold based on the Minimum Description Length principle (Rissanen, 1978). Despite those, there also exist approaches that focus on reducing the bias of estimating MI (Vinh et al., 2014; Mandros et al., 2017) or the Shannon entropy (Paninski, 2003; Han et al., 2015; Valiant and Valiant, 2011), or different data types. The latter, we discuss in the next chapter.

In this thesis, we motivate our approach by considering algorithmic CMI based on Kolmogorov complexity (Kolmogorov, 1965), which we introduce in Section 3.1. In contrast to the empirical estimator for CMI, algorithmic CMI does not only consider the empirical distribution, but also its complexity, which leads to a more robust estimation. Although Kolmogorov complexity is not computable, for discrete data, we can approximate it from above via the minimax optimal normalized maximum likelihood (NML) (Shtarkov, 1987), to which we state the definition in Section 3.2. Further, we discuss factorized NML and quotient NML, which are both variations of NML for conditional distributions. In Section 3.3, we show how to approximate algorithmic CMI via different versions of NML and further prove that the corresponding estimator, SCCI, is a strongly consistent estimator of CMI. Next, in Section 3.4, we discuss the relation between SCCI and the corrections proposed by Goebel et al. (2005) and Suzuki (2016). After that, we empirically analyze the sample complexity of SCCI in Section 3.5. As our experiments suggest, SCCI needs a sub-linear amount of samples to detect independencies. Last, we empirically benchmark our test against state-of-the-art conditional independence tests for discrete data and show that it improves the accuracy of constraint-based discovery algorithms in Section 3.6. For reproducibility, we provide our test in the SCCI R-package.

3.1 Conditional Independence Testing

In this section, we introduce the notation and give brief introductions to both conditional independence testing using conditional mutual information (CMI), as well as to the notion of algorithmic conditional independence, which is defined through Kolmogorov complexity.

Since in this part, we do not focus on differences between separations and independencies, as we assume that faithfulness and the causal Markov condition hold, we slightly abuse the notation and denote $\perp\!\!\!\perp_P$ with $\perp\!\!\!\perp$, whenever clear from context.[1] Under this setup, our goal is to test the conditional independence hypothesis $H_0\colon X \perp\!\!\!\perp Y \mid Z$ against the general alternative $H_1\colon X \not\perp\!\!\!\perp Y \mid Z$, where X, Y and Z denote possibly multivariate discrete random vectors with finite sample spaces (or domain sizes) $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$. In the context of independence testing in causal graphs, the random variables X and Y would be univariate in most cases, while Z could refer to a set of random variables. To not make the notation overly complicated, we will only use the set notation, e.g. $\boldsymbol{Z}$, when we want to clarify that we are referring to random variables that represent multiple nodes in a causal graph.

A perfect independence test minimizes both the type I error, that is, falsely rejecting the null hypothesis, as well as the type II error—i.e., falsely accepting the null hypothesis. A high type I error will lead to finding spurious edges in a causal discovery setup while having a high type II error means that we will miss out on true edges. A well-known measure for conditional independence is *conditional mutual information* (CMI) based on *Shannon entropy* (Cover and Thomas, 2012). The Shannon entropy of a possibly multivariate discrete random variable X with probability mass p is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \ ,$$

and the conditional Shannon entropy of a discrete random variable X given Y is defined as

$$H(X \mid Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(y)} \ .$$

Using these definitions, we can define conditional mutual information as

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Z, Y) \ ,$$

where $X \perp\!\!\!\perp Y \mid Z$ iff $I(X;Y \mid Z) = 0$.

If we are given the true distribution of a random variable, CMI is ideal to test for conditional independencies on discrete data. In practice, we need to work with a limited sample size. On such a limited sample the plug-in estimator $\hat{H}$ tends to underestimate conditional entropies, and as a consequence, conditional mutual information is overestimated—even for completely independent data, as the following example shows.

[1] Note that assuming faithfulness, every independence testing implies a d-separation.

Example 3.1 *Given three random variables X_1, X_2 and Y, with respective domain sizes $1\,000, 8$ and 4. Suppose that we are given 500 samples drawn from their joint distribution and find that $\hat{H}(Y \mid X_1) = \hat{H}(Y \mid X_2) = 0$. That is, Y is a deterministic function of X_1, as well as of X_2. However, as $|\mathcal{X}_1| = 1\,000$, and given only 500 samples, it is likely that a large fraction of values $v \in \mathcal{X}_1$ is only assigned to a single data point. Thus, finding that $\hat{H}(Y \mid X_1) = 0$ is likely due to the limited amount of samples, rather than that it depicts a true (functional) dependency, while $\hat{H}(Y \mid X_2) = 0$ is more likely to be due to a true dependency, since the number of samples $n \gg |\mathcal{Y}| \times |\mathcal{X}_2|$—i.e., the support for each value in the sample space is $\gg 1$.*

A possible solution is to set a threshold t such that $X \perp\!\!\!\perp Y \mid Z$ if $\hat{I}(X;Y \mid Z) \leq t$. Setting t is, however, not an easy task, as t is dependent on the quality of the entropy estimate, which by itself strongly depends on the complexity of the distribution and the given number of samples. Instead, to avoid this problem altogether, we will base our test on the notion of *algorithmic* independence.

3.1.1 Algorithmic Independence

To define algorithmic independence, we first need to briefly introduce to Kolmogorov complexity (Kolmogorov, 1965; Li and Vitányi, 2019).

Definition 3.1 ((Prefix) Kolmogorov Complexity) *The (prefix) Kolmogorov complexity of a finite binary string x is the length of the shortest self-delimiting binary program p^* for a universal prefix Turing machine $\mathcal{U}$ that generates x, and then halts. Formally, we have*

$$K(x) = \min\{|p| \mid p \in \{0,1\}^*, \mathcal{U}(p) = x\} \; .$$

In other words, program p^* is the most succinct *algorithmic* description of x, or the ultimate lossless compressor for that string. Importantly, the above definition defines prefix Kolmogorov complexity. There also exists plain Kolmogorov complexity, for which the program does not have to be self-delimiting, that is, it is assumed that the machine knows where codewords start and end. In this chapter, this distinction does not make a difference, however, for consistency throughout the thesis, we use prefix Kolmogorov complexity.

To define algorithmic independence, we will also need conditional (prefix) Kolmogorov complexity, that is,

$$K(x \mid y) = \min\{|q| \mid q \in \{0,1\}^*, \mathcal{U}(y,q) = x\} \; .$$

In essence, we still try to find the shortest program that outputs x, but we get y as an additional input. Similar to Shannon entropy, providing more information can only reduce the length of the program and hence $K(x \mid y) \leq K(x) + \mathcal{O}(1)$, where the constant does not depend on x or y. It is common to avoid writing the additional constant complexity term and instead write $\overset{+}{\leq}$ or $\overset{\pm}{=}$ to indicate that the inequality respectively the equality holds up to an additive constant.

By definition, Kolmogorov complexity makes maximal use of any effective structure in x; structure that can be expressed more succinctly algorithmically than by printing it verbatim. As such it is the theoretical optimal measure for complexity. In the context of our problem, instead of purely considering the statistical dependence between random variables, it also considers the complexity of the generating mechanism that induces the dependence.

Let us consider Example 3.1 again and let x_1, x_2, and y be the binary strings representing of the samples drawn from X_1, X_2 and Y. As Y can be expressed as a deterministic function of X_1 or X_2, $K(y \mid x_1)$ and $K(y \mid x_2)$ reduce to the programs describing the corresponding function. As the domain size of X_2 is 8 and $|\mathcal{Y}| = 4$, the program that describes the function from X_2 to Y only has to describe the mapping from 8 to 4 values, which will be shorter than describing a mapping from X_1 to Y, since $|\mathcal{X}_1| = 1\,000$—i.e., $K(y \mid x_2) \overset{+}{\leq} K(y \mid x_1)$ in contrast $\hat{H}(Y \mid X_1) = \hat{H}(Y \mid X_2)$.

To reject $X \perp\!\!\!\perp Y \mid Z$, we test whether providing information about Y and Z leads to a shorter program than only knowing Z (Chaitin, 1975).

Definition 3.2 (Algorithmic CMI) *Given the strings x, y and z, we write z^* to denote the shortest program for z, and analogously $(z, y)^*$ for the shortest program for the concatenation of z and y. Algorithmic conditional mutual information is defined as*

$$I_A(x; y \mid z) = K(x \mid z^*) - K(x \mid (z, y)^*) \; .$$

Similar to CMI, we say that x is algorithmically independent of y given z iff $I_A(x; y \mid z) \overset{\pm}{=} 0$. Due to the halting problem Kolmogorov complexity is, however, not computable nor approximable up to arbitrary precision (Li and Vitányi, 2019). The Minimum Description Length (MDL) principle (Grünwald, 2007) provides a statistically well-founded approach to approximate Kolmogorov complexity from above.

3.1.2 MDL a Brief Primer

The Minimum Description Length (MDL) principle (Grünwald, 2007; Rissanen, 1978) is a practical variant of Kolmogorov Complexity. Simply put, instead of all programs, it considers only those programs that we know output x *and* halt. Intuitively, given a collection of models, also called a model class $\mathcal{M}$, MDL identifies the model $M^* \in \mathcal{M}$, which compresses the given data D best. Formally, our goal is to find

$$M^* = \operatorname*{argmin}_{M \in \mathcal{M}} L(D|M) + L(M),$$

where $L(M)$ is the length in bits needed to describe the model M or identify M within the model class $\mathcal{M}$, and $L(D \mid M)$ is the length in bits of the description of data D given M. Such a model class can be defined very generally. As we will see below, a model class could for example be the set of all parametric distributions with a parameter vector θ consisting of k entries. We could even specify this class further and only consider multinomials. Another example of a model class, as we will see in the next chapter, can be the set of all possible discretizations of a random variable w.r.t. a given precision ϵ.

The general concept of splitting up the encoding into the code length of the data given the model and the code length of the model is also known as *two-part MDL*. There also exists one-part, or *refined* MDL, where we encode data w.r.t. to whole model class. Refined MDL is superior as it avoids arbitrary choices in the description language L, but in practice it is only computable for certain model classes, such as multinomials. Given infinite data the model costs degenerate to an additive logarithmic term, which is independent of the data. Hence given infinite data, two-part MDL behaves similar to refined MDL. Note that in either case we are only concerned with code *lengths*—our goal is to measure the *complexity* of a dataset under a model class, not to actually compress it (Grünwald, 2007).

In the following section, we will introduce a refined MDL code for multinomials and use it to approximate algorithmic CMI.

3.2 Stochastic Complexity for Multinomials

In the following, we will define stochastic complexity for multinomials, which belongs to the class of refined MDL codes. On a high level, we will define stochastic complexity as the negative logarithm of the *normalized maximum likelihood* (NML), which has several nice properties.

Let X be a discrete random variable with $|\mathcal{X}| = k$, where we assume that X can be modelled by a parametric distribution P_θ, with parameter vector $\theta = (\theta_1, \ldots, \theta_k)$. Further, we denote all distributions that can be described with such a k-dimensional parameter θ by $\mathcal{M}_k$. Given a sample x^n of n data points drawn w.r.t. P_θ we denote the maximum likelihood (ML) estimate of θ w.r.t. to x^n by $\hat{\theta}(x^n)$. Shtarkov (1987) defined the NML density function as

$$f_{NML}(X \mid \mathcal{M}_k) = \frac{f_{\hat{\theta}(x^n)}(x^n)}{\mathcal{C}^n_{\mathcal{M}_k}} \ , \tag{3.1}$$

where $f_{\hat{\theta}(x^n)}$ is the empirical density function for X based on the maximum likelihood estimate $\hat{\theta}(x^n)$ under the model class $\mathcal{M}_k$. The normalizing factor, or regret $\mathcal{C}^n_{\mathcal{M}_k}$, relative to the model class $\mathcal{M}_k$ is defined as

$$\mathcal{C}^n_{\mathcal{M}_k} = \sum_{\tilde{x}^n \in \mathcal{X}^n} f_{\hat{\theta}(\tilde{x}^n)}(\tilde{x}^n) \,.$$

The sum iterates over every possible sample $\tilde{x}^n$ of length n and sample space $\mathcal{X}$, and for each considers the ML estimate for that data given model class $\mathcal{M}_k$. Whenever clear from context, we will drop $\mathcal{M}_k$ to simplify the notation—i.e., we write $P_{NML}(x^n)$ for $P_{NML}(x^n \mid \mathcal{M}_k)$ and let $\mathcal{C}^n_k$ to refer to $\mathcal{C}^n_{\mathcal{M}_k}$.

Notably, as shown by Rissanen (2001) the NML distribution incorporates all information in the data that can be extracted with the models in the model class $\mathcal{M}_k$. Moreover, the NML distribution is the optimal encoding w.r.t. the model class even if the data was generated by a model outside $\mathcal{M}_k$. The latter was formally shown by Rissanen (2001), who proved that besides solving Shtarkov's minimax problem (Shtarkov, 1987), the NML distribution is also the solution to the minimax problem

$$\inf_q \sup_g E_g \log \frac{f_{\hat{\theta}(x^n)}(x^n)}{q(x^n)} \,,$$

where the distributions q and g can range over virtually any distribution—i.e., g can be a distribution outside the model class. Since in this thesis, we only use NML as an encoding, we refer the reader that is interested in the optimality properties of NML to Rissanen (2001).

Important for us is that for discrete data, we can assume the model class to be the class of multinomial distributions. Under this assumption, we can rewrite Equation (3.1) as (Kontkanen and Myllymäki, 2007)

$$f_{NML}(x^n) = \frac{\prod_{j=1}^{k} \left(\frac{c_j}{n}\right)^{c_j}}{\mathcal{C}^n_k} \,,$$

where c_j is the empirical frequency of the j-th value in the sample space $\mathcal{X}$ in x^n. Respectively we can compute the regret as

$$\mathcal{C}^n_k = \sum_{c_1+\cdots+c_k=n} \frac{n!}{c_1!\cdots c_k!} \prod_{j=1}^{k} \left(\frac{c_j}{n}\right)^{c_j} \,.$$

Fortunately, Mononen and Myllymäki (2008) derived a formula to calculate the regret in sub-linear time, meaning that the whole formula can be computed in linear time w.r.t. n.

Building upon the definition of f_{NML}, we obtain the *stochastic complexity* of a discrete random variable X based on a sample x^n by simply taking the negative logarithm[2]—i.e.,

$$\begin{aligned} \mathrm{SC}(X) &= -\log f_{NML}(x^n) \ , \\ &= n\hat{H}(X) + \log \mathcal{C}_k^n \ . \end{aligned}$$

As a result, we see that the stochastic complexity decomposes into n times the empirical entropy and the log regret, which is also called *parametric complexity.*

3.2.1 Conditional Stochastic Complexity

Conditional stochastic complexity can be defined in different ways. We consider factorized normalized maximum likelihood (fNML) (Silander et al., 2008) and quotient normalized maximum likelihood (qNML) (Silander et al., 2018), which are equivalent except for the regret terms.

Given an empirical sample over two random vectors X and Y, conditional stochastic complexity using fNML is defined as

$$\begin{aligned} \mathrm{SC}_f(X \mid Y) &= -\sum_{y \in \mathcal{Y}} \log f_{NML}(x^n \mid y^n = y) \\ &= n\hat{H}(X \mid Y) + \sum_{y \in \mathcal{Y}} \log \mathcal{C}_{|\mathcal{X}|}^{c_y} \ , \end{aligned}$$

where c_y corresponds to the number of samples for which $Y = y$. Analogously, we define conditional stochastic complexity using qNML (Silander et al., 2018)

$$\begin{aligned} \mathrm{SC}_q(X \mid Y) &= -\log \frac{f_{NML}(x^n, y^n)}{f_{NML}(y^n)} \\ &= n\hat{H}(X \mid Y) + \log \frac{\mathcal{C}_{|\mathcal{X}| \cdot |\mathcal{Y}|}^n}{\mathcal{C}_{|\mathcal{Y}|}^n} \ . \end{aligned}$$

In the following, we refer to conditional stochastic complexity as SC and only use SC_f or SC_q whenever there is a conceptual difference. Further, we denote to the regret term of $\mathrm{SC}(X)$ as $\mathcal{R}(X) = \log \mathcal{C}_{|\mathcal{X}|}^n$ and respectively refer

[2] As is common for MDL encodings, we want to obtain a code-length in terms of bits and hence compute the logarithm with respect to basis 2 and define $0 \log 0 = 0$.

to the regret of $\mathrm{SC}(X \mid Y)$ as $\mathcal{R}(X \mid Y)$, where

$$\mathcal{R}_f(X \mid Y) = \sum_{y \in \mathcal{Y}} \log \mathcal{C}^{c_y}_{|\mathcal{X}|} \textit{ and}$$

$$\mathcal{R}_q(X \mid Y) = \log \frac{\mathcal{C}^n_{|\mathcal{X}| \cdot |\mathcal{Y}|}}{\mathcal{C}^n_{|\mathcal{Y}|}} \; .$$

Next, we show that $\mathcal{C}^n_k$ is log-concave in n, which is a property we need to guarantee that our estimator is always smaller or equal than the empirical estimator $\hat{I}(X;Y \mid Z)$.

Lemma 3.1 *For $n \geq 1$, the regret $\mathcal{C}^n_k$ of the multinomial stochastic complexity of a random variable with a domain size of $k \geq 2$ is log-concave in n.*

For readability, we postpone the proof of Lemma 3.1 to Appendix A.4. In the following theorem, we present the first implication of this Lemma.

Theorem 3.1 *Given three discrete random variables X, Y and Z with domain sizes ≥ 2, it holds that $\mathcal{R}(X \mid Z) \leq \mathcal{R}(X \mid Z, Y)$.*

Proof: *We start by proving the statement for $\mathcal{R}_f$. Consider that Z contains p distinct value combinations $\{r_1, \dots, r_p\}$. If we add Y to Z, the number of distinct value combinations, $\{l_1, \dots, l_q\}$, increases to q, where $p \leq q$. Consequently, to show the claim, it suffices to show that*

$$\sum_{i=1}^{p} \log \mathcal{C}^{c_i}_k \leq \sum_{j=1}^{q} \log \mathcal{C}^{c_j}_k \tag{3.2}$$

where $\sum_{i=1}^{p} c_i = \sum_{j=1}^{q} c_j = n$. Next, consider w.l.o.g. that each value combination $\{r_i\}_{i=1,\dots,p}$ is mapped to one or more value combinations in $\{l_1, \dots, l_q\}$. Hence, Equation (3.2) holds, if $\log \mathcal{C}^n_k$ is sub-additive in n. Since we know from Lemma 3.1 that the regret term is log-concave in n (since both $p, q \geq 2$), sub-additivity follows by definition.

Next, consider $\mathcal{R}_q$. Let k, p and q be the domain sizes of X, Y and Z, we need to show that

$$\mathcal{R}_q(X \mid Z) \leq \mathcal{R}_q(X \mid Z, Y)$$
$$\Leftrightarrow \log \frac{\mathcal{C}^n_{kq}}{\mathcal{C}^n_q} \leq \log \frac{\mathcal{C}^n_{kpq}}{\mathcal{C}^n_{pq}} \; .$$

We know from Silander et al. (2018) that for $p \in \mathbb{N}, p \geq 2$, the function $q \mapsto \frac{\mathcal{C}^n_{p \cdot q}}{\mathcal{C}^n_q}$ is increasing for every $q \geq 2$. This suffices to prove the statement above. □

3.3 Stochastic Complexity based Conditional Independence

With the above, we can formulate our new conditional independence test, which we will refer to as the ***S**tochastic **C**omplexity based **C**onditional **I**ndependence criterium*, or SCCI.

Definition 3.3 (SCCI) *Let X, Y and Z be discrete random vectors, SCCI is defined as*

$$\begin{aligned} \mathit{SCCI}(X;Y \mid Z) &= \mathit{SC}(X \mid Z) - \mathit{SC}(X \mid Z,Y) \\ &= n\hat{I}(X;Y \mid Z) + \mathcal{R}(X \mid Z) - \mathcal{R}(X \mid Z,Y)\ . \end{aligned} \tag{3.3}$$

Further, we say that $X \perp\!\!\!\perp Y \mid Z$ if $\mathit{SCCI}(X;Y \mid Z) \leq 0$.

From the second row in Equation 3.3, we see that the regret terms formulate a natural threshold t_{SC} for the empirical estimate of CMI, where $t_{\mathrm{SC}} = \mathcal{R}(X \mid Z,Y) - \mathcal{R}(X \mid Z)$. From Theorem 3.1 we know that if we instantiate SCCI using fNML or qNML, we are guaranteed that $\mathcal{R}(X \mid Z,Y) - \mathcal{R}(X \mid Z) \geq 0$. Hence, Y has to provide a significant gain such that SCCI$(X;Y \mid Z) > 0$—i.e., we need $\hat{H}(X \mid Z) - \hat{H}(X \mid Z,Y) > t_{\mathrm{SC}}/n$. In other words, if

$$\hat{I}(X;Y \mid Z) \leq \frac{t_{\mathrm{SC}}}{n}\ ,$$

we would consider X and Y to be independent given Z. Thus, it is obvious that no matter what formulation of conditional stochastic complexity we choose, SCCI is more restrictive than the empirical estimator of CMI.

3.3.1 Factorized SCCI

To formulate our independence test based on factorized normalized maximum likelihood, we have to revisit the regret terms again. In particular, $\mathcal{R}_f(X \mid Z)$ is only equal to $\mathcal{R}_f(Y \mid Z)$, when the domain size of X is equal to the domain of Y. Further, $\mathcal{R}_f(X \mid Z) - \mathcal{R}_f(X \mid Z,Y)$ is not guaranteed to be equal to

$\mathcal{R}_f(Y \mid Z) - \mathcal{R}_f(Y \mid Z, X)$. Consequently, our test would not be symmetric. Hence, we formulate SCCI using fNML as

$$\begin{aligned}\mathrm{SCCI}_f(X;Y \mid Z) &= n\hat{I}(X;Y \mid Z) \\ &\quad + \max\{\mathcal{R}(X \mid Z) - \mathcal{R}(X \mid Z,Y), \mathcal{R}(Y \mid Z) - \mathcal{R}(Y \mid Z,X)\}\ .\end{aligned}$$

An alternative way to obtain a symmetric test using fNML would be to base the test on an equivalent formulation of (algorithmic) CMI, that is

$$\begin{aligned}I_A(x,y \mid z) &= K(x \mid z^*) - K(x \mid (z,y)^*) \\ &\overset{+}{=} K(x \mid z^*) + K(y \mid z^*) - K((x,y) \mid z^*)\ . \end{aligned} \tag{3.4}$$

If we approximate this alternative formulation using fNML, we get

$$\mathrm{SCCI}_{fs}(X;Y \mid Z) = \mathrm{SC}_f(X \mid Z) + \mathrm{SC}_f(Y \mid Z) - \mathrm{SC}_f(X,Y \mid Z)\ .$$

By writing down the regret terms, we see that SCCI_{fs} is symmetric. In particular, if we only consider the regret terms, we get

$$\sum_{z \in Z} \left(\mathcal{C}^{c_z}_{|\mathcal{X}|} + \mathcal{C}^{c_z}_{|\mathcal{Y}|} - \mathcal{C}^{c_z}_{|\mathcal{X}||\mathcal{Y}|} \right)\ .$$

All regret terms depend on the factorization given Z. For the previous formulation, however, we compare the factorizations of X given only Z to the one given Z and Y, or respectively the factorization of Y given only Z to the one given Z and X. Thus, for SCCI_f all regret terms correspond to the same domain, either to the domain of X or Y, whereas for SCCI_{fs} the regret terms are based on X, Y and the Cartesian product of them. Due to the latter, SCCI_{fs} is more conservative than SCCI_f, as we will show in our experiments. Apart from the fact that SCCI_f is more robust in the high-dimensional setup, both variants have a similar performance, which is why we mainly consider SCCI_f in the experiments.

3.3.2 Quotient SCCI

To formulate SCCI using quotient normalized maximum likelihood, we can straightforwardly replace SC with SC_q in the independence criterium—i.e.

$$\mathrm{SCCI}_q(X;Y \mid Z) = \mathrm{SC}_q(X \mid Z) - \mathrm{SC}_q(X \mid Z,Y)\ .$$

By writing down the regret terms for $\mathrm{SCCI}_q(X;Y \mid Z)$ and $\mathrm{SCCI}_q(Y;X \mid Z)$, we can see that they are equal and hence SCCI_q is symmetric. Another nice property of the qNML formulation is that we would get an equivalent formulation, if we were to base SCCI_q on the alternative formulation of algorithmic (CMI) that we showed in Equation 3.4. The only shortcoming of this formulation is that similar to SCCI_{fs}, SCCI_q is more restrictive than SCCI_f and thus does not perform as well on high-dimensional data.

Another way to instantiate SCCI, is to use the asymptotic approximation of stochastic complexity (Rissanen, 1996), which was done by Suzuki (2016) to approximate CMI. In practice, the corresponding test (JIC) is, however, very restrictive, which leads to a low recall.

Next, we will show that SCCI is a consistent estimator of CMI and hence in the sample limit able to reliably distinguish (conditional) independencies from dependencies. Thereafter, we compare SCCI to CMI using the threshold based on the gamma distribution (Goebel et al., 2005), and empirically evaluate the sample complexity of SCCI on a limited sample.

3.3.3 SCCI as a Consistent Estimator of CMI

In this part, our goal is to show that $\frac{1}{n}\mathrm{SCCI}$ approaches the true conditional mutual information, as $n \to \infty$. That is, we need to show that

$$\frac{1}{n}\mathrm{SCCI} = \hat{I} + \frac{t_{\mathrm{SC}}}{n}$$

approaches CMI. Since it is known that $\hat{I} \to I$ when $n \to \infty$ almost surely (Antos and Kontoyiannis, 2001), it remains to prove that t_{SC}/n approaches zero, which we will show below.

Theorem 3.2 *Given three discrete random variables X, Y and Z, we have that $\lim_{n\to\infty} \frac{1}{n}$SCCI$(X;Y \mid Z) = I(X;Y \mid Z)$, almost surely.*

Proof: *To show the claim, we need to show that*

$$\lim_{n\to\infty} \hat{I}(X;Y \mid Z) + \frac{1}{n}(\mathcal{R}(X \mid Z) - \mathcal{R}(X \mid Z,Y)) = I(X;Y \mid Z) \ .$$

The proof for any alternative formulation of SCCI follows equivalently. Since it is known that $\hat{I} \to I$ when $n \to \infty$ almost surely (Antos and Kontoyiannis, 2001), we need to show that $\frac{1}{n}(\mathcal{R}(X \mid Z) - \mathcal{R}(X \mid Z,Y))$ goes to zero as n goes to infinity. From Rissanen (1996) we know that $\log \mathcal{C}_k^n$ asymptotically behaves like $\frac{k-1}{2}\log n + \mathcal{O}(1)$, i.e., only grows logarithmically w.r.t. n. Hence, $\frac{1}{n}\mathcal{R}(X \mid Z)$ and $\frac{1}{n}\mathcal{R}(X \mid Z,Y)$ will approach zero if $n \to \infty$. □

3.4 Link to Related Estimators

Goebel et al. (2005) estimate conditional mutual information through a second-order Taylor series and show that their estimator can be approximated with the gamma distribution. In particular, they state that

$$\hat{I}(X;Y \mid Z) \sim \Gamma\left(\frac{|\mathcal{Z}|}{2}(|\mathcal{X}|-1)(|\mathcal{Y}|-1), \frac{1}{n \ln 2}\right) ,$$

where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ refer to the domains of X, Y and Z. This means by selecting a significance threshold α, we can derive a threshold for CMI based on the gamma distribution—for convenience we call this threshold t_Γ. In the following, we compare t_Γ against $t_{\mathrm{SC}} = \mathcal{R}(X \mid Z, Y) - \mathcal{R}(X \mid Z)$.

First of all, for qNML, like t_Γ, t_{SC} depends purely on the sample size and the domain sizes. However, we consider the difference in complexity between only conditioning X on Z and the complexity of conditioning X on Z and Y. For fNML, we have the additional aspect that the regret terms for both $\mathcal{R}(X \mid Z)$ and $\mathcal{R}(X \mid Z, Y)$ also relate to the probability mass function of Z, and respectively the Cartesian product of Z and Y. Recall that for k being the size of the domain of X, we have that

$$\mathcal{R}_f(X \mid Z) = \sum_{z \in Z} \log \mathcal{C}_k^{c_z} .$$

As $\mathcal{C}_k^n$ is log-concave in n (Lemma 3.1), $\mathcal{R}_f(X \mid Z)$ is maximal if Z is uniformly distributed—i.e., it is maximal when $H(Z)$ is maximal. This is a favourable property, as the probability that Z is equal to X is minimal for uniform Z, as stated in the following Lemma.

Lemma 3.2 (Cover and Thomas (2012)) *If X and Y are i.i.d. with entropy $H(Y)$, then $P(Y = X) \geq 2^{-H(Y)}$ with equality if and only if Y has a uniform distribution.*

To elaborate on the link between t_Γ and t_{SC}, we compare them empirically. In addition, we compare the results to the threshold provided from the JIC test. First, we compare t_Γ with $\alpha = 0.05$ and $\alpha = 0.001$ to t_{SC}/n for fNML, qNML, and JIC on fixed domain sizes, with $|\mathcal{X}|$=$|\mathcal{Y}|$=$|\mathcal{Z}|$=4 and varying sample sizes (see Figure 3.1). For fNML we computed the worst case threshold by modelling Z as uniformly distributed. In general, the behaviour for each threshold is similar, whereas qNML, fNML and JIC are more restrictive than t_Γ.

Next, we keep the sample size fixed at 500 and increase the domain size of Z from 2 to 200, to simulate a multivariate random vector. Except to JIC, which seems to overpenalize in this case, we observe that fNML is most restrictive until we reach a plateau when $|\mathcal{Z}| = 125$. This is due to the fact

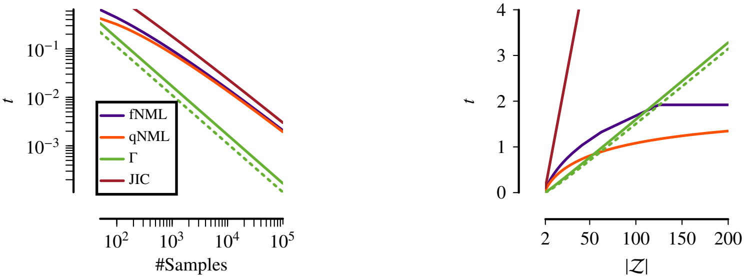


Figure 3.1: Threshold for CMI using fNML, qNML, JIC and the gamma distribution with $\alpha = 0.05$ (solid) and $\alpha = 0.001$ (dashed) for different sample sizes and fixed domain sizes equal to four (left) and fixed sample size of 500 and changing domain sizes (right).

that $|\mathcal{Z}||\mathcal{Y}| = 500$ and hence each data point is assigned to one value in the Cartesian product. We have that $\mathcal{R}_f(X \mid Z, Y) = |\mathcal{Z}||\mathcal{Y}|\mathcal{C}^1_k$.

Note that for the thresholds that we computed for fNML we pretend that Z and Y are divided equally over the joint domain $|\mathcal{Y}||\mathcal{Z}|$. In practice, this requirement may not be fulfilled, and hence the regret term for fNML can be smaller. In addition, it is possible that the number of distinct value combinations for Y and Z that we observe in the sample is smaller than their Cartesian product, which also reduces the regret for the fNML formulation.

3.5 Empirical Sample Complexity

In this section, we empirically evaluate the sample complexity of SCCI_f, where we focus on the type I error, i.e., $H_0\colon X \perp\!\!\!\perp Y \mid Z$ is true and hence $I(X;Y \mid Z) = 0$. We generate data accordingly and draw samples from the joint distribution, where we set $P(x,y,z) = \frac{1}{|\mathcal{X}||\mathcal{Y}||\mathcal{Z}|}$ for each value configuration $(x,y,z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Per sample size we draw $1\,000$ data sets and report the average absolute error for SCCI_f and the empirical estimator of CMI. We show the results for two cases in Figure 3.2. We observe that in contrast to the empirical estimator $\hat{I}$, SCCI_f quickly approaches zero, and that the difference between both estimators is especially large if we increase the sample space.

If we take a second look at those plots, we see that SCCI_f only needs 180 samples to reach an average error smaller than 0.05 (in both), while the size of the domain space for the second experiment is $|\mathcal{X}||\mathcal{Y}||\mathcal{Z}| = 256$. Ideally, we would like to compute the number of samples that is needed for a reliable result as a function of the domain sizes of the involved variables. Formally, we would like to know the number of samples n that is required such that

$$P(|\mathrm{SCCI}_f(X;Y \mid Z)/n - I(X;Y \mid Z)| \geq \epsilon) \leq \delta \; .$$

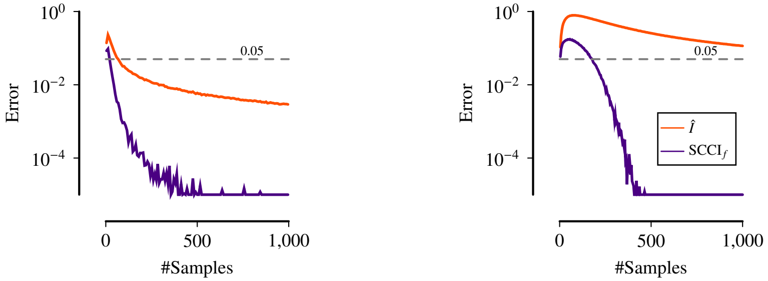


Figure 3.2: Error for SCCI_f and $\hat{I}$ compared to I, where $I(X;Y|Z) = 0$. Left: $|\mathcal{X}| = |\mathcal{Y}| = 4$ and $|\mathcal{Z}| = 4$. Right: $|\mathcal{X}| = |\mathcal{Y}| = 4$ and $|\mathcal{Z}| = 16$. Values smaller than 10^{-5} are truncated to 10^{-5}.

As a theoretical analysis is very challenging, we try to derive an empirical bound for $\epsilon = \delta = 0.05$.

We generate data according to the independence hypothesis like above and conduct empirical evaluations for varying domain sizes of X, Y and Z, where we define w.l.o.g. $|\mathcal{X}| \geq |\mathcal{Y}|$, as the test is symmetric. For each combination of domain sizes, we compute $P(|\mathrm{SCCI}_f(X;Y \mid Z)/n - I(X;Y \mid Z)| \geq \epsilon) = P(\mathrm{SCCI}_f(X;Y \mid Z)/n \geq 0.05) \leq 0.05$ as follows: We start with a small n, e.g. two, generate 1 000 data sets and check if over those data sets $P(\mathrm{SCCI}_f(X;Y \mid Z)/n \geq 0.05) \leq 0.05$ holds. If not, we increase n by the minimum domain size of X, Y and Z. We repeat this procedure until we reach an n, for which $P(\mathrm{SCCI}_f(X;Y \mid Z)/n \geq 0.05) \leq 0.05$ holds and report this n.

In Figure 3.3 we plot those values for varying either the domain sizes of X, Y or Z independently or jointly. From these evaluations, we handcrafted a formula to show that it is possible to find an n that is sub-linear w.r.t. the domain sizes of X, Y and Z for which empirically $P(\mathrm{SCCI}_f(X;Y \mid Z)/n \geq 0.05) \leq 0.05$ always holds. Hence, we additionally plot for each domain size the corresponding suggested bound for the sample complexity w.r.t. the formula $35 + 2|\mathcal{X}||\mathcal{Y}|^{\frac{2}{3}}(|\mathcal{Z}| + 1)$. We observe that the empirical values for n are always smaller than the values provided by this formula. Despite this positive result, we want to emphasize that this is only an example function to show the existence of a sub-linear bound for this data. From the plots we would expect that there exists an even tighter bound, however, we did not optimize for that. For future work we would like to theoretically validate a sub-linear bound function.

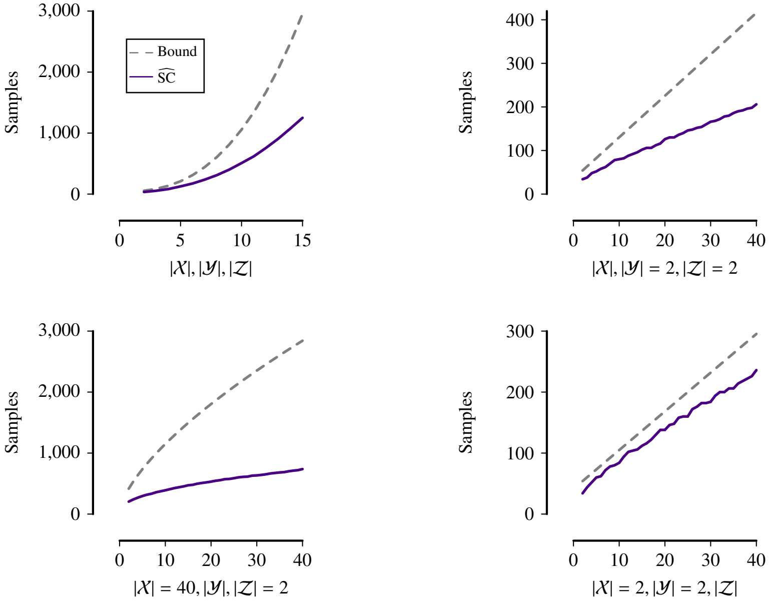


Figure 3.3: Estimated sample complexities for independently generated data s.t. $P(|\mathrm{SCCI}_f/n - I| \geq 0.05) \leq 0.05$. The suggested bound is calculated as $35 + 2|\mathcal{X}||\mathcal{Y}|^{\frac{2}{3}}(|\mathcal{Z}| + 1)$. For all setups, increasing the domain size of X, Y, Z together or independently, the bound function is larger than the empirical value.

3.6 Experiments

In this section, we empirically evaluate SCCI based on fNML and compare it to the alternative formulation using qNML. To not overload the plots, we postpone most comparisons to SCCI_{fs} to Appendix A.5. In addition, we compare our results to the G^2 test from the *pcalg* R package (Kalisch et al., 2012), CMI_Γ (Goebel et al., 2005) and JIC (Suzuki, 2016).

3.6.1 Identifying Conditional (In)dependencies

To test whether SCCI can reliably identify (in)dependencies, we generate data according to the graph shown in Figure 3.4, where we assign the values of F uniformly w.r.t. to its domain space and model a dependency from X to Y by uniformly assigning a mapping form X to Y. We set the domain size for each variable to four and generate data under various samples sizes (100–2 500) and additive uniform noise settings (0%–95%). For each setup we generate 200 data sets and assess the accuracy. We report the correct identifications of $F \perp\!\!\!\perp T \mid$

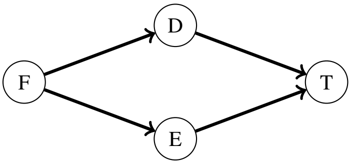


Figure 3.4: [d-Separation] Given the above causal DAG, F is d-separated from T given D, E and hence by CMC it holds that $F \perp\!\!\!\perp T \mid \{D, E\}$. Assuming that faithfulness holds, we additionally get that $D \not\!\perp\!\!\!\perp T \mid \{E, F\}$ as well as $E \not\!\perp\!\!\!\perp T \mid \{D, F\}$.

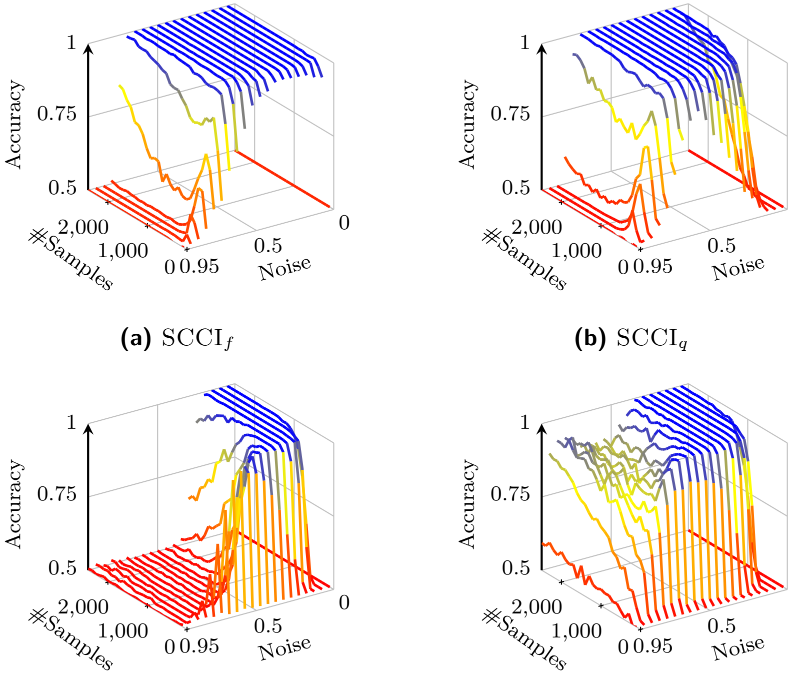
$\{D, E\}$ as the true positive rate and the false identifications $D \perp\!\!\!\perp T \mid \{E, F\}$ or $E \perp\!\!\!\perp T \mid \{D, F\}$ as false positive rate.[3] For the G^2 test and CMI_Γ we select $\alpha = 0.05$, however, we found no significant differences for $\alpha = 0.01$.

We show the accuracy of the best performing competitors in Figure 3.5 and report the remaining results as well as the true and false positive rates for each approach in Appendix A.5. Overall, we observe that SCCI_f performs near perfect for less than 70% noise, while for $\geq$ 70% additive noise, the type II error increases. Those results are even better than expected as from our empirical bound function we would suggest that at least 378 samples are required to have reliable results for this data set. SCCI_q has a similar but slightly worse performance. In contrast, CMI_Γ only performs well for less than 30% noise and fails to identify true independencies after more than 30% noise has been added, which leads to a high type I error. The G^2 test has problems with sample sizes up to 500 and performs inconsistently for more than 35% noise.

3.6.2 Changing the Domain Size

Using the same data generator as above, we now consider a different setup. We fix the sample size to 2 000 and use only 10% additive noise—a setup in which all tests performed well. What we change is the domain size of the source F from 2 to 20 while also restricting the domain sizes of the remaining variable to the same size. For each setup we generate 200 data sets.

From the results in Figure 3.6 we can clearly see that only SCCI_f performs well for larger domain sizes, whereas all other test have a false positive rate of 100% for $|\mathcal{F}| > 10$, resulting in an accuracy of 50%.



(a) SCCI_f

(b) SCCI_q

(c) CMI_Γ

(d) G^2

Figure 3.5: [Higher is better] Accuracy of SCCI_f, SCCI_q, CMI_Γ and G^2 for identifying d-separation using varying samples sizes and additive noise percentages, where a noise level of 0.95 refers to 95% additive noise. Note that for 0% noise the relation is deterministic.

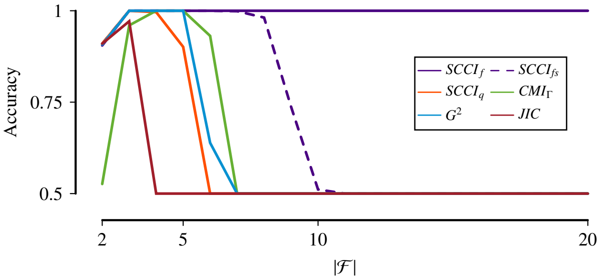


Figure 3.6: D-separation example with $2\,000$ samples and 10% noise. We gradually increase the domain size of the source node F, which propagates through the graph.

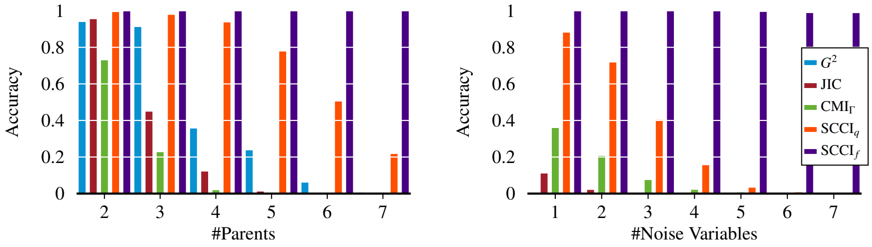


Figure 3.7: Left: Percentage of parents identified, where we start with only two parents and increase the number of parents to seven. Right: Percentage of parents identified, where we always use three parents and add independently generated noise variables to the conditioning set.

3.6.3 Identifying Multiple Parents

In this experiment, we test the type II error. This we do by generating a certain number of parents Pa_T from which we generate a target node T. To generate the parents, we use either a

- uniform distribution with domain size $d \sim \mathrm{Unif}(2, 5)$,
- geometric distribution with parameter $p \sim \mathrm{Unif}(0.6, 0.8)$,
- hyper-geometric distribution with parameter $K \sim \mathrm{Unif}(4, 6)$, or
- Poisson distribution with parameter $\lambda \sim \mathrm{Unif}(1, 2)$.

Given Pa_T, we generate T as a mapping from the Cartesian product of the parents to T plus 10% additive uniform noise. Then we generate for each distribution 200 data sets with 2 000 samples, per number of parents $k \in \{2, \ldots, 7\}$. We apply SCCI_f, SCCI_q, CMI_Γ and G^2 on each data set and check $\forall P \in \mathrm{Pa}_T$ if the corresponding test correctly identifies that $P \not\perp T \mid \mathrm{Pa}_T \setminus \{P\}$.

We plot the averaged results for each k in Figure 3.7. It can clearly be observed that SCCI_f performs best and still has near to 100% accuracy for seven parents. Although not plotted here, we can add that the competitors struggled most with the data drawn from the Poisson distribution. We assume that this is due to the fact that the domain sizes for these data sets were on average larger than for the remaining distributions.

In the next experiment, we generate parents and target in the same way, whereas we now fix the number of parents to three. In addition, we generate $k \in \{1, \ldots, 7\}$ random variables N that are drawn jointly independent from T and Pa_T and are uniformly distributed. Then we assess whether the tests under consideration can still identify for each $P \in \mathrm{Pa}_T$ that $P \not\perp T \mid N \cup \mathrm{Pa}_T \setminus \{P\}$.

[3] For 0% noise all functions are deterministic, which leads to a faithfulness violation and thus $D \not\perp T \mid \{E, F\}$ and $E \not\perp T \mid \{D, F\}$ does not hold. Consequently, an accuracy of 50% is the best we can hope for in this setting.

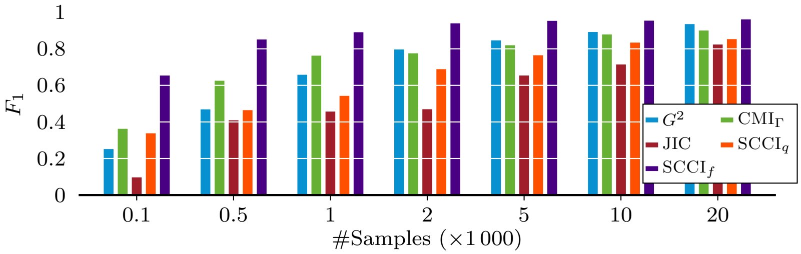


Figure 3.8: [Higher is better] F_1 score on undirected edges for stable PC using SCCI_f, SCCI_q, JIC, CMI_Γ and G^2 on the *Alarm* network for different sample sizes.

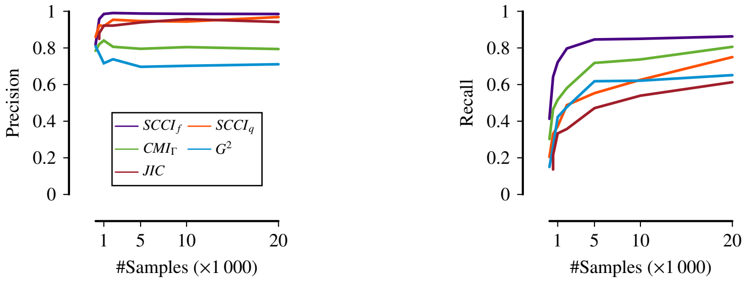


Figure 3.9: [Higher is better] Precision (left) and recall (right) for PCMB using SCCI_f, SCCI_q, JIC, CMI_Γ and G^2 to identify all Markov blankets in the *Alarm* network for different sample sizes.

The averaged results for G^2, JIC, SCCI_f, SCCI_q and CMI_Γ are plotted in Figure 3.7. Notice that the results for G^2 are barely visible, as they are close to zero for each setup. In general, the trend that we observe is similar to the previous experiment, except that the differences between SCCI_f and its competitors are even larger.

3.6.4 Causal Discovery with SCCI

Last, we evaluate how SCCI performs in practice. Thus, we run the stable PC algorithm (Kalisch et al., 2012; Colombo and Maathuis, 2014) on the *Alarm* network (Scutari and Denis, 2014) from which we generate data with different sample sizes and average over the results of ten runs for each sample size. We equip the stable PC algorithm with SCCI_f, SCCI_q, JIC, CMI_Γ and the default, the G^2 test, and plot the average F_1 score over the undirected graphs in Figure 3.8. We observe that our proposed test, SCCI_f outperforms its competitors by a large margin, especially for $n \leq 1\,000$.

As a second practical test, we compute the Markov blanket for each node in the *Alarm* network and report the precision and recall. To find the Markov blankets, we run the PCMB algorithm (Peña et al., 2007) with the four independence tests. We plot the precision and recall for each variant in Figure 3.9. We observe that again SCCI_f performs best—especially with regard to recall. As for Markov blankets of size m it is necessary to condition on at least $m-1$ variables, the advantage in recall can be linked back to SCCI_f being able to correctly detect dependencies for larger domain sizes.

3.7 Conclusion

In this chapter, we introduced SCCI, a conditional independence test for discrete data. We derived SCCI from algorithmic conditional independence and showed how to instantiate it using different variants of NML. In the sample limit, we showed that SCCI is a strongly consistent estimator of CMI, while given only a few samples, it outperforms state-of-the-art independence tests. To examine the efficiency of SCCI further, we empirically analyzed its sample complexity for detecting independencies. Our analysis suggests that a sublinear amount of samples is sufficient to find independencies, however, formally proving this result is out of the scope of this thesis. Since not all interesting data is discrete, we consider the more general case in the next chapter.

Chapter 4

Towards Independence Testing on Mixture Data

In the previous chapter, we showed how to estimate conditional mutual information on discrete data, however, not all data is discrete. In many-real world settings, the data may be continuous, may concern a mix of discrete and continuous random variables, such as age (in years) and height, or even random variables that can individually consist of a *mixture* of discrete and continuous components. For the latter, consider the photoelectric effect, where electrons are emitted after electromagnetic radiation, such as light, hits a surface. Below a certain level of radiation, no reaction is happening, while after this threshold, the reaction is put into motion.[1] Hence, if we were to measure the joint distribution of both quantities, the variable quantifying the rate of the reaction would be such a mixture variable.

While estimating (conditional) mutual information for purely discrete or continuous data is a well-studied problem (Cover and Thomas, 2012; Darbellay and Vajda, 1999; Gao et al., 2016; Han et al., 2015; Paninski and Yajima, 2008), not much work has focused on mixture random variables. Although there exist several discretization-based methods that can estimate MI for a mix of discrete and continuous random variables (Cabeli et al., 2020; Mandros et al., 2020; Suzuki, 2016), so far, only methods based on k-nearest neighbor (kNN) estimation were shown to work on *mixed variables*, which may consist of discrete-continuous mixture variables (Gao et al., 2017; Mesner and Shalizi,

This chapter is based on Marx, Yang, and van Leeuwen (2021b).

[1] Thanks to Lukas Klemmer for bringing up this example.

2020; Rahimzamani et al., 2018).

Regardless of the success of kNN-based estimators, discretization-based approaches have attractive properties, e.g., with regard to global interpretation. That is, a natural and understandable way to discretize a continuous random variable is via creating a histogram model, where we cut the sample space of the continuous variable into multiple non-overlapping parts called bins (Scott, 2015), or (hyper)rectangles for multi-dimensional variables. Within a bin, we consider the distribution to be constant, which allows us to estimate the density function via Riemann integration by making the bins smaller and smaller (Cover and Thomas, 2012). This definition, however, is less straightforward when discrete-continuous mixture variables are involved.

We approach this problem as follows: we first extend the definition of entropy for a univariate discrete-continuous mixture variable given by Politis (1991) to multivariate variables. Using this definition, we show that CMI for mixed random variables can be written as a sum of entropies that are well-defined through the Radon-Nikodym derivate (see Section 4.1). Exploiting this property, we propose a consistent CMI estimator for such data that is based on adaptive histogram models in Section 4.2. To efficiently learn adaptive histograms from data, in Section 4.3, we define a model selection criterion based on the Minimum Description Length principle (Rissanen, 1978). Subsequently, we propose an iterative greedy algorithm that aims to obtain the histogram model that minimizes the proposed MDL score in Section 4.4. We discuss related work in Section 4.5, and in Section 4.6, we empirically show that our method performs favorably to state-of-the-art estimators for mixed data and can be used in a causal discovery setting.

4.1 Entropy for Mixed Random Variables

Formally, we consider multi-dimensional *mixed random variables*, of which any individual dimension can be discrete, continuous, or a discrete-continuous mixture. Further, we call a vector of such mixed random variables a *mixed random vector*. For a mixed random vector (X, Y), where X and Y are possibly multivariate, we cannot use the same definition of entropy or mutual information, as in the previous section. Instead, we need to adopt the most general definition of mutual information (MI), i.e., the measure-theoretic definition:

$$I(X;Y) = \int_{\mathcal{X}\times\mathcal{Y}} \log \frac{dP_{XY}}{dP_X P_Y} dP_{XY} \ ,$$

where $dP_{XY}/(dP_X P_Y)$ is the *Radon-Nikodym derivative*, dP_{XY} the joint measure, and $P_X P_Y$ the product measure. It has been proven that $P_X P_Y$ is *absolutely continuous* with respect to P_{XY} (Gao et al., 2017), i.e., $P_{XY} = 0$

whenever $P_X P_Y = 0$; and therefore, such a Radon-Nikodym derivative always exists and $I(X, Y)$ is well-defined. This measure-theoretic definition can be extended to CMI using the chain rule: $I(X; Y \mid Z) = I(X; \{Y, Z\}) - I(X; Z)$.

As we saw in the previous chapter, CMI for purely discrete data can be written as a sum of (conditional) entropies, e.g. $I(X; Y \mid Z) = H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z)$. The same holds for purely continuous data. What is not clear, however, is if this formula also holds when (X, Y, Z) contains discrete-continuous mixture random variables. We investigate this problem in two steps. We first define the measure-theoretic entropy for a (possibly multi-dimensional) discrete-continuous mixture random variable and prove it to be well-defined, though previous work claimed the opposite (Gao et al., 2017). Second, using this definition, we prove that (conditional) MI for a mixed random vector can be written as the sum of measure-theoretic entropies, just like purely continuous or discrete random vectors.

4.1.1 A Generalized Definition of Entropy

The measure-theoretic entropy is defined only for univariate random variables (Politis, 1991). For this definition, we give an explicit proof that such a univariate measure-theoretic entropy is well-defined, and then extend its definition to the multi-dimensional case, which we prove is also well-defined.

Generalized One-Dimensional Entropy

We start off by reviewing the existing definition for the one-dimensional case (Politis, 1991). Given a one-dimensional random variable X

$$H(X) = \int_{\mathbb{R}} \frac{dP_X(x)}{dv(x)} \log \frac{dP_X(x)}{dv(x)} dv(x),$$

where $v(\cdot)$ is a measure defined on all one-dimensional Borel sets (Politis, 1991). If $v(\cdot)$ is the Lebesgue measure, which we denote as $u(\cdot)$, $H(X)$ becomes the differential entropy. Alternatively, if $v(\cdot)$ is a counting measure, $H(X)$ becomes the common (discrete) entropy.

If, however, X is a discrete-continuous mixture variable, v is defined as follows. We split $\mathbb{R}$ into three disjoint subsets s.t. $\mathbb{R} = S_d \cup S_c \cup S_o$. First, S_o is the subset of $\mathbb{R}$ on which X has zero probability measure, i.e., $P_X(S_o) = 0$. Second, the set S_d contains all discrete points, i.e., S_d is countable and $\forall x \in S_d, P_X(x) > 0$. Third, S_c covers the continuous points, hence $P_X(S_c) + P_X(S_d) = 1$ and for any Borel set $A \subseteq S_c$ satisfying $u(A) = 0$, we have $P_X(A) = 0$. Based on these three subsets S_d, S_c, and S_o, we can define v as

$$v(A) = u(A \cap S_c) + |A \cap S_d| \,, \tag{4.1}$$

where $|A \cap S_d|$ is the cardinality of this intersection.

To show that the generalized one-dimensional entropy is well-defined, we need to prove that the Radon-Nikodym derivative dP_X/dv always exists. This we show in the following lemma.

Lemma 4.1 *Given a one-dimensional discrete-continuous random variable X with probability measure P_X, P_X is absolutely continuous w.r.t. v, i.e., $P_X = 0$ whenever $v = 0$, and hence dP_X/dv always exists.*

PROOF: *Given a Borel set $A \subseteq \mathbb{R}$ such that $v(A) = u(A \cap S_c) + |A \cap S_d| = 0$, we have $u(A \cap S_c) = 0$ due to non-negativity of any measure, as well as $|A \cap S_d| = 0$. Since $A \cap S_c \subseteq S_c$, by the definition of S_c we have $P(A \cap S_c) = 0$. It remains to show that $A \cap S_d = \emptyset$, which we do by contradiction. Assume that $A \cap S_d \neq \emptyset$, then there exists $x \in A \cap S_d$ s.t. for a set containing only x, $|\{x\}| = 1$. Then $|A \cap S_d| \geq |\{x\}| = 1$, which contradicts $|A \cap S_d| = 0$. Thus, we must have $A \cap S_d = \emptyset$ and then $P_X(A) = 0$.* □

Next, we extend this definition to multi-dimensional variables.

Generalized Multi-Dimensional Entropy

In the following, we extend the measure-theoretic entropy definition to a mixed m-dimensional random vector $W = (W_1, \ldots, W_m)$, where each W_i is a one-dimensional variable. For each W_i, we define S_d^i, S_c^i, S_o^i and measure v^i as above, and also define the *product measure* v for the m-dimensional random vector as $v = v^1 \times \ldots \times v^m$. Then, define the entropy for W as

$$H(W) = \int_{\mathbb{R}^m} \frac{dP_W(w)}{dv(w)} \log \frac{dP_W(w)}{dv(w)} dv(w). \tag{4.2}$$

To prove that such entropy is well-defined, we show that dP_W/dv always exists.

Lemma 4.2 *Given a mixed m-dimensional random vector $W = (W_1, \ldots, W_m)$ with probability measure P_W, dP_W/dv always exists.*

PROOF: *Given a m-dimensional Borel set A, there exist one-dimensional Borel sets $A_1, \ldots, A_m$ such that $A = A_1 \times \ldots \times A_m$. If $v(A) = 0$, then there exists at least one $v^i, i \in \{1, \ldots, m\}$, such that $v^i(A_i) = 0$. Thus, by Lemma 4.1, $P_{W_i}(A_i) = 0 \Rightarrow P_W(\mathbb{R} \times \ldots \times \mathbb{R} \times A_i \times \mathbb{R} \times \ldots \times \mathbb{R}) = 0 \Rightarrow P_W(A) = 0$, as $A = A_1 \times \ldots \times A_m \subseteq \mathbb{R} \times \ldots \times \mathbb{R} \times A_i \times \mathbb{R} \times \ldots \times \mathbb{R}$.* □

Last, based on Lemma 4.1 and 4.2, we can prove that just like for a purely continuous or discrete random vector, conditional mutual information for a mixed random vector can be written as a sum of entropies.

Lemma 4.3 *Given a mixed random vector (X, Y, Z) with joint probability measure P_{XYZ}, we can write $I(X; Y \mid Z) = H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z)$, where each entropy can be defined as in Eq.* (4.2).

PROOF: *We first prove the statement for $Z \neq \emptyset$, for which we can write $I(X;Y|Z) = I(X; \{Y, Z\}) - I(X; Z)$ by the chain rule for mutual information. Thus, it suffices to prove that $I(X; Z) = H(X) + H(Z) - H(X, Z)$ and $I(X; \{Y, Z\}) = H(X) + H(Y, Z) - H(X, Y, Z)$. Next, denote v as the product measure defined based on (X, Z), where $v = v^1 \times \ldots \times v^{m_{XZ}}$, and m_{XZ} is the number of dimensions of X plus that of Z; then by Lemma 4.2, we also have $P_{XZ} \ll v$. Next, we show that $P_X P_Z \ll v$. For some m_{XZ}-dimensional Borel set $A = A_1 \times \ldots \times A_{m_{XZ}}$ satisfying $v(A) = 0$ there exists $v^i \in \{v^1, \ldots, v^{m_{XZ}}\}$ such that $v^i(A_i) = 0$. Hence, $P_X P_Z(A) = 0$ because $0 \leq P_X P_Z(A) = P_X P_Z(A_1 \times \ldots \times A_{m_{XZ}}) \leq P_X P_Z(\mathbb{R} \times \ldots \mathbb{R} \times A_i \times \mathbb{R} \ldots \times \mathbb{R}) = P_i(A_i) = 0$, where P_i is the marginalization of the product measure $P_X P_Z$ to the ith dimension and $P_i(A_i) = 0$ is because $v^i(A_i) = 0$ by the definition of v.*

Finally, by the chain rule of the Radon-Nikodym derivative we have that

$$\begin{aligned} I(X; Z) &= \int \log \frac{dP_{XZ}}{dP_X P_Z} dP_{XZ} \\ &= \int \log \frac{dP_{XZ}/dv}{dP_X P_Z/dv} (dP_{XZ}/dv) dv \\ &= H(X) + H(Z) - H(X, Z) \; . \end{aligned}$$

The proof for $I(X; \{Y, Z\})$ is equivalent. If $Z = \emptyset$, CMI reduces to $I(X; Y)$, for which we can prove the statement in the same manner. □

As a direct implication of the above proof, it follows that mutual information can also be written as the sum of entropies, since it is a special case of CMI with $Z = \emptyset$. With this generalized definition, we can now show how to estimate CMI using adaptive histogram models.

4.2 Adaptive Histogram Models

Adaptive histogram models have been thoroughly studied for continuous random variables (Scott, 2015); however, to the best of our knowledge, there exists no rigorous definition of histograms for mixed random variables. Thus, to use histogram models as a foundation to estimate the measure-theoretic (conditional) MI, we need to rigorously define histograms for mixed random variables. We start with the one-dimensional case.

4.2.1 One-Dimensional Histogram Models

A histogram model is typically defined based on a set of consecutive intervals called *bins* (Scott, 2015). However, to deal with discrete-continuous mixture random variables, we define the set of bins, denoted as B, such that each bin is either an interval or a set containing only a single point. That is, $B = B' \cup B''$, where B' and B'' are sets of subsets of $\mathbb{R}$, with B' consisting of countable consecutive intervals and B'' consisting of countable single point sets. Last, we define the "width" of a bin using the measure v as defined in Equation 4.1, i.e., for a bin $B_j \in B$ we have

$$v(B_j) = u(B_j \cap B') + |B_j \cap B''| \,.$$

As any $B_j \in B''$ contains only a single discrete point, $v(B_j) = 1$ for all $B_j \in B''$.

Further, we define a histogram model M as a set of bins equipped with a parameter vector of length k, where $k = |B|$ is the number of bins. That is, a histogram model M is a family of probability distributions $P_{X,\theta}$, parametrized by the vector $\theta = (\theta_1, \ldots, \theta_k)$. Each element of θ represents the Radon-Nikodym derivative (or density) of each bin. Note that this definition generalizes to purely continuous random variables when $B'' = \emptyset$ and also to discrete random variables if $B' = \emptyset$. For the latter case, the histogram model degenerates to a multinomial model.

4.2.2 Multi-Dimensional Histograms

For a mixed m-dimensional random vector $W = (W_1, \ldots, W_m)$, we define the set of *bins* for each W_i as in Section 4.2.1, denoted as B^i. Consequently, we can define a set of m-dimensional *bins*, denoted B, by the Cartesian product $B = B^1 \times \ldots \times B^m$. Since each B^i is countable, B is also countable, and we can hence assume B is indexed by j. Then, we split B in a similar way as in the one-dimensional case, i.e., $B = B' \cup B''$, where B'' contains *only discrete values.* That is, for any m-dimensional bin $B_j \in B''$, each dimension of B_j is a set that contains a single one-dimensional point. Note that, however, for any $B_j \in B'$, each dimension of B_j can either be a one-dimensional interval or a one-dimensional single-point set. Further, we define the volume of a multi-dimensional bin $B_j \in B$ using the product measure $v(B_j)$ (see Section 4.1.1).

Similar to univariate histograms, a multi-dimensional histogram model M can be described by a probability distribution $P_{W,\theta}$ parametrized by the vector

$\theta = (\theta_1, \ldots, \theta_k)$, where k is the number of bins and θ_i is the Radon-Nikodym derivative for each bin.

4.2.3 Maximum Likelihood Estimator

Given a possibly multi-dimensional histogram with k bins, we denote the Radon-Nikodym derivative $dP_{W,\theta}/dv$ as f^h_θ and its MLE as $f^h_{\hat{\theta}}$. Observe that for any parameter $\theta_j \in \theta$, the product $\theta_j v(B_j)$ follows a multinomial distribution. Thus, given a dataset $D = \{D_i\}_{i=1,\ldots,n}$, with D_i representing a row, the maximum log-likelihood is denoted as and equal to

$$l_M(D) = \log f^h_{\hat{\theta}(D)}(D) = \log \prod_{j=1}^{k} \left(\frac{c_j}{n \cdot v(B_j)} \right)^{c_j}, \tag{4.3}$$

where c_j and $v(B_j)$ are respectively the number of data points and the bin volumes of bin $j \in \{1 \ldots k\}$. Notice that this maximum likelihood generalizes to the discrete case (i.e., multinomial distribution) when all $v(B_j) = 1$, and to the continuous case (Scott, 2015) when v becomes the Lebesgue measure.

4.2.4 Conditional Mutual Information Estimator

Combining all previous theoretical discussions, we can estimate conditional mutual information for (possibly multivariate) random variables X, Y, Z by

$$I^h(X; Y \mid Z) = H^h(X, Z) + H^h(Y, Z) - H^h(X, Y, Z) - H^h(Z) .$$

The corresponding measure-theoretic entropies are estimated from m-dimensional data over (X, Y, Z), where m_X, m_Y and m_Z are the corresponding number of dimensions of X, Y and Z. We estimate the entropies as

$$\begin{aligned}
H^h(X, Y, Z) &= -\int_{\mathbb{R}^m} f^h_{\hat{\theta}}(x, y, z) \log(f^h_{\hat{\theta}}(x, y, z)) dv \\
H^h(X, Z) &= -\int_{\mathbb{R}^{m_X + m_Z}} f^h_{\hat{\theta}}(x, z) \log(f^h_{\hat{\theta}}(x, z)) dv \\
H^h(Y, Z) &= -\int_{\mathbb{R}^{m_Y + m_Z}} f^h_{\hat{\theta}}(y, z) \log(f^h_{\hat{\theta}}(y, z)) dv \\
H^h(Z) &= -\int_{\mathbb{R}^{m_Z}} f^h_{\hat{\theta}}(z) \log(f^h_{\hat{\theta}}(z)) dv
\end{aligned}$$

in which $f^h_{\hat{\theta}}(x, y, z)$ is the maximum likelihood estimator given the data, while we obtain $f^h_{\hat{\theta}}(x, z)$, $f^h_{\hat{\theta}}(y, z)$, and $f^h_{\hat{\theta}}(z)$ via marginalization from $f^h_{\hat{\theta}}(x, y, z)$.

Next, we will prove that I^h is a strongly consistent estimator for conditional mutual information on mixed data.

Theorem 4.1 *Given a mixed random vector (X, Y, Z) with probability measure P_{XYZ}, $\lim_{v' \to 0} \lim_{n \to \infty} I^h(X; Y \mid Z) = I(X; Y \mid Z)$ almost surely, where n refers to the sample size and v' refers to the maximum of the histogram volumes for bins in B' (defined in Section 4.2.2).*

The proof is provided in Appendix A.6. Informally, our proof is based on the following key aspects: 1) All volume-related terms in I^h cancel out, 2) discrete empirical entropy converges to the true entropy almost surely (Antos and Kontoyiannis, 2001), and 3) in the limit, differential entropy can be obtained by discretizing a continuous random variable into "infinitely" small bins (Cover and Thomas, 2012, Theorem 8.3.1). Notably, the order of the double limit in Theorem 4.1 inherently indicates that n should grow faster than the number of bins (Rudin, 1964), which is also required for histograms on purely continuous data to converge (Scott, 2015).

4.3 Learning Adaptive Histograms from Data

To efficiently estimate a histogram model that inherits the consistency guarantees from Theorem 4.1 we need to consider the following requirements. First of all, we need to ensure that we learn a joint histogram model over (X, Y, Z). This is due to the fact that we obtain the lower-dimensional entropies such as $H^h(X, Z)$ by marginalization over the maximum likelihood estimator $f^h_{\hat{\theta}}(x, y, z)$. If we would not learn a joint model, the volume-related terms in $H^h(X, Y, Z)$, $H^h(X, Z)$, $H^h(Y, Z)$, and $H^h(Z)$ would not cancel out. In addition, we need to make sure that the number of bins is in $o(n)$—i.e., grows sub-linear w.r.t. n, while at the same time the size of the bins decreases.

One way to achieve those properties would be to fix the bin width or the number of bins depending on the number of samples. However, such an approach is not very flexible and does not allow for variable bin widths. To allow for a more flexible model, we formally consider the problem of constructing an adaptive multi-dimensional histogram as a model selection problem and employ a selection criterion based on MDL (see Section 3.1.2). MDL-based model selection has been successfully used for learning one-dimensional (Kontkanen and Myllymäki, 2007) and two-dimensional histograms (Kameya, 2011; Yang et al., 2020), demonstrating adaptivity to both local density changes and sample size.

We now briefly define the MDL-optimal histogram model. Specifically, while previous work (Kameya, 2011; Kontkanen and Myllymäki, 2007; Yang et al., 2020) only considers purely continuous data (or more precisely, data with arbitrarily small precision), we apply the MDL principle to mixed type data, based on our rigorous definition of histogram models for mixed random variables. On top of that, we empirically show that our score fulfils the desired properties, that is, the number of bins grows as $o(n)$.

4.3.1 MDL Histograms

To encode a histogram model, we resort to two-part MDL. Hence, we first need to define a model class $\mathcal{M}$, then the encoding of a model $M \in \mathcal{M}$ and last the encoding of the data given a model M.

Given a dataset D with n rows and m individual columns D^j, we now define the model class $\mathcal{M}$. First, we create fixed bins according to B'' (as defined in Section 4.2.2) per discrete value that occurs in D_j. Next, we enumerate all possible bins for B' with fixed precision ϵ to encode the continuous part of D_j. To this end, denote the remaining non-discrete data points in D_j as D^c_j. If D^c_j is empty D_j corresponds to a discrete variable and we can stop here. Otherwise, we create all possible cut points for D^c_j as

$$C^0_j = \{\min(D^c_j), \min(D^c_j) + \epsilon, \dots, \max(D^c_j)\} \; .$$

By selecting a subset of cut points $C_j \subseteq C^0_j$, we get a valid solution for B'. We can enumerate all possible segmentations by enumerating each $C_j \subseteq C^0_j$. By repeating this process for each dimension, we obtain our model class $\mathcal{M}$. Further, we get the code length for a model $M \in \mathcal{M}$ by encoding all combinations of cut points for each dimension (Kontkanen and Myllymäki, 2007), i.e.,

$$L(M) = \sum_{j \in \{1,\dots,m\}} L(C_j) = \sum_{j \in \{1,\dots,m\}} \log \binom{|C^0_j|}{|C_j|} \; .$$

If we take a step back, we notice that such a model M in essence describes a multivariate discrete variable. If we denote k as the size of the domain space of the cartesian product of that multivariate random variable, we can encode it via NML, as in the last chapter—i.e.,

$$L(D \mid M) = -l_M(D) + \log \mathcal{C}^n_k \; ,$$

where $-l_M(D)$ is the maximum log-likelihood of data D under a histogram model M (see Equation 4.3) and $\log \mathcal{C}^n_k$ is the parametric regret as defined in Section 3.2. In other words, we encode the data given the model with the stochastic complexity. Thus, we get the total code length $L(M, D) = L(D \mid M) + L(M)$ of our two-part MDL code.

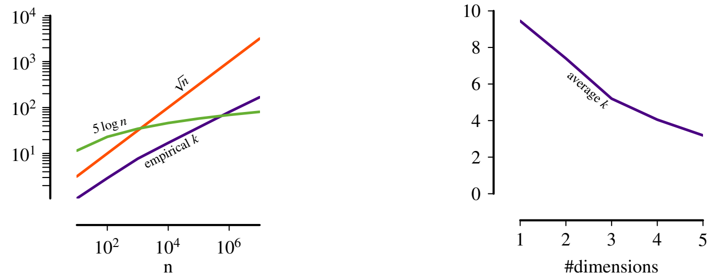


Figure 4.1: Left: Average number of bins k to discretize $X \sim N(0,1)$ for increasing sample sizes (20 repetitions per sample size). Right: Per dimension of a multivariate Gaussian distribution with $X_i \perp\!\!\!\perp X_j$ and $X_i \sim N(0,1)$, we show the average number of bins ($n = 2\,000$, 20 repetitions).

4.3.2 Empirical Consistency of the MDL Score

To prove consistency for our score, we need to show that the number of selected bins grows at rate $o(n)$. Since the theoretical analysis is rather difficult, we instead empirically demonstrate this property for one-dimensional Gaussian distributions. As we show in Figure 4.1, the average number of bins k obtained for a Gaussian distribution, grows with n, but slower than $\sqrt{n}$. In addition, for multi-dimensional data, for which we can only approximate the histogram model that minimizes $L(D, M)$, we observe that if the number of dimensions increases, the average number of bins per dimension decreases if we keep n fixed. Thus, there is empirical evidence that our score has this desired property.

Next, we explain how we compute those discretizations.

4.4 Implementation

In the following, we describe our algorithm to estimate the joint entropy for an m-dimensional discrete-continuous mixture random vector $X = (X_1, \ldots, X_m)$.

4.4.1 Algorithm

To discretize a one-dimensional random variable X, we first create bins for the discrete values of X and then discretize the continuous values. We detect discrete data points by checking if a single value x in the domain $\mathcal{X}$ of X occurs multiple times. If a user-defined threshold t, e.g. 5 is reached, we create a bin for this point. To discretize the remaining continuous values, we start by splitting $\mathcal{X}$ into K_{init} equi-width bins, which we can safely choose from the complexity class $\mathcal{O}(\sqrt{n})$ as we saw in the previous section. Using dynamic

Algorithm 4.1:

```
  input  : Data $D = \{D_1, \ldots, D_m\} \sim X$;
           Maximum number of iterations $i_{\max}$
  output: Discretization $X_d$
1 $X_d \leftarrow \text{init}(D)$; $i \leftarrow 1$;
2 while $X_d$ changes $\wedge\ i \leq i_{\max}$ do
3     $X_d^i \leftarrow X_d$;
4     foreach $j \in \{1, \ldots, m\}$ do
5         $X_d^{ij} \leftarrow \text{refine}(D_j \mid X_d)$;
6         if $\text{SC}(X_d^{ij}) < \text{SC}(X_d^i)$ then
7             $X_d^i \leftarrow X_d^{ij}$;
8     $X_d \leftarrow X_d^i$; $i \leftarrow i + 1$;
9 return $X_d$;
```

programming, we compute the variable-width histogram model M that minimizes $L(D, M)$ in quadratic time w.r.t. K_{init}, as proposed by Kontkanen and Myllymäki (2007).

Since the runtime complexity to compute the optimal variable-width histogram over a multi-dimensional random variable would grow exponentially w.r.t. m, we opt for an iterative greedy algorithm, for which we provide the pseudocode in Algorithm 4.1. As input, we are given a dataset $D = \{D^1, \ldots, D^m\}$ consisting of n rows and m columns, representing a sample of size n from an m-dimensional random vector X, and a user-specified parameter $i_{\max}$ specifying the maximum number of iterations. First, we initialize the discretization X_d (line 1) by creating single bin histograms for the continuous points in D_j and a bin with bin-width 1 per discrete point. To detect the latter, we check if there exist $|\{x \in \mathcal{X}_j \mid D_j = x\}| \geq t$, where t is a user-defined threshold. After that, we iteratively update the discretization for that X_j providing the highest gain in stochastic complexity, until either the score cannot be improved or the maximum number of iterations has been reached (lines 2–8). To update the discretization of a variable X_j we call the function refine (line 5), which receives as input the data D_j and the discretization after iteration i. It then re-discretizes X_j using an extension of the dynamic programming algorithm by Kontkanen and Myllymäki (2007). In essence, instead of simply discretizing X_j independently of the remaining variables, we keep the discretizations for all $X_i \neq X_j$ fixed and find the optimal histogram for X_j s.t. the overall score $L(D, M)$ is minimized.

4.4.2 Complexity

The complexity of discretizing a univariate random variable is in $\mathcal{O}(K_{\max} \cdot (K_{\text{init}})^2)$ and depends on the number of initial bins K_{init} and the maximum number of bins $K_{\max}$, which we typically chose as a fraction of K_{init} (both in $o(\sqrt{n})$). In a multi-dimensional setting we have to multiply this complexity by the current domain size of the remaining variables, since we have to update each bin conditioned on those. In the worst case, this number is equal to $(K_{\max})^{m-1}$. Overall, we apply this procedure—if all variables are continuous—$i_{\max} \cdot m$ times.

4.5 Related Work

Here, we discuss related methods for adaptive histograms and (conditional) mutual information estimation for different data types.

Both theoretical properties and practical issues of density estimation using histograms have been studied for decades (Scott, 2015). Various algorithms have been proposed for the challenging task of constructing an adaptive one-dimensional histogram, among which the MDL-based histogram (Kontkanen and Myllymäki, 2007) is considered to be the state-of-the-art, as it is self-adaptive to both local density structure and sample size; and does not have any hyperparameters. Learning adaptive multivariate histograms is even harder due to the combinatorial explosion of the search space. One approach is to resort to the dyadic CART algorithm (Klemelä, 2009); various methods designed for specific tasks also exist (Kameya, 2011; Weiler and Eggert, 2007). Our algorithm is similar to that of Kameya (2011), but they only consider the two-dimensional case.

For estimating (conditional) mutual information on continuous data or a mix of discrete and continuous data, three classes of approaches exist. The first class concerns kernel density estimation (KDE) methods (Gao et al., 2016; Paninski and Yajima, 2008), which perform well on continuous data; however, no KDE-based MI and CMI estimation method exists that is designed for discrete-continuous mixture random variables. Moreover, bandwidth tuning for KDE can be extensive and computationally expensive, which becomes even worse when the data is not purely continuous, as different bandwidths may be needed when discrete random variables take different values. The second class of methods relies on k-nearest neighbor (kNN) estimates (Frenzel and Pompe, 2007; Kozachenko and Leonenko, 1987; Kraskov et al., 2004), which have been established as the state-of-the-art (Gao et al., 2017; Rahimzamani et al., 2018). kNN approaches can be applied not only to a mix of discrete and continuous variables, but can also be used as consistent MI (Gao et al., 2017) and CMI (Mesner and Shalizi, 2020; Rahimzamani et al., 2018) estimators for discrete-continuous mixtures. The third class of methods first discretizes the continuous random variables and then calculates the mutual information from the discretized variables (Cover and Thomas, 2012; Darbellay and Vajda, 1999; Suzuki, 2016). Two recent approaches based on adaptive partitioning for data that consists of discrete and continuous variables have been proposed (Cabeli et al., 2020; Mandros et al., 2020). While Mandros et al. (2020) focus on mutual information and its application to functional dependency discovery, Cabeli et al. (2020), similar to us, build upon an MDL-based score to estimate MI and CMI, to which we compare in Section 4.6. The key difference is that Cabeli et al. (2020) compute $I(X;Y \mid Z)$ as $(I(X;\{Y,Z\}) - I(X;Z) + I(Y;\{X,Z\}) - I(Y;Z))/2$ and maximize each of the four terms (with penalty terms) directly, while we first learn a joint histogram.

To the best of knowledge, we are the first to propose a CMI estimator for discrete-continuous mixture variables based on discretization or histogram density estimation. Our method can consistently estimate CMI on mixed random variables containing discrete-continuous mixtures. We focus on histogram-based models instead of kNN estimation, since histograms are more interpretable (Scott, 2015) and do not require tuning of the parameter k, which can have a large impact on the outcome.

4.6 Experiments

In this section, we empirically evaluate the performance of our approach. First, we will benchmark our estimator against state-of-the-art CMI estimators on different data types. After that, we evaluate how well our estimator is suited to test for conditional independence in a causal discovery setup. For reproducibility, we make our code available online.[2]

4.6.1 Mutual Information Estimation

On the mutual information estimation task, we compare our approach to the state-of-the-art MI estimators. In particular, we compare against FP (Frenzel and Pompe, 2007), RAVK (Rahimzamani et al., 2018) and MS (Mesner and Shalizi, 2020), which all rely on kNN estimates, and MIIC (Cabeli et al., 2020), which is a discretization-based method. All of those can be applied to our setup, but only the authors of RAVK and MS specifically consider discrete-continuous mixture variables. We apply MIIC using the default parameters and use $k = 10$ for all kNN-based approaches.[3] For our algorithm, we set the

[2] `https://github.com/ylincen/CMI-adaptive-hist.git`

[3] We evaluated all kNN estimators with $k = 5, 10, 20$. Since $k = 10$ had the best trade-off, we report those results.

maximum number of iterations and the threshold to detect discrete points in a mixture variable to 5, set $K_{\text{init}} = 20 \log n$ and $K_{\max} = 5 \log n$. To comply with the literature, we compute all entropies using the *natural logarithm.*

Experiment I-IV

As a sanity check, we start with an experiment on purely continuous data. That is, for **Experiment I**, let X and Y be Gaussian distributed random variables with mean 0, variance 1, and covariance 0.6. Consequently, the correlation ρ between X and Y is 0.6 and the true MI can be calculated as $I(X;Y) = -\frac{1}{2}\log(1-\rho^2)$. In **Experiment II**, X is discrete and drawn from $\text{Unif}(0, l-1)$, with $l = 5$ and Y is continuous with $Y \sim \text{Unif}(x, x+2)$ for $X = x$. Therefore, $I(X;Y) = \log(l) - \frac{(l-1)\log 2}{l}$ (Gao et al., 2017). Next, for **Experiment III**, X is exponentially distributed with rate 1 and Y is a zero-inflated Poissonization of X—i.e., $Y = 0$ with probability $p = 0.15$ and $Y \sim \text{Pois}(x)$ for $X = x$ with probability $1-p$. The ground truth is $I(X;Y) = (1-p)(2\log 2 - \gamma - \sum_{k=1}^{\infty} \log k \cdot 2^{-k}) \approx (1-p)0.3012$, where γ is the Euler-Mascheroni constant (Gao et al., 2017). Last, in **Experiment IV**, we generate the data according to the Markov chain $X \to Z \to Y$ (see Mesner and Shalizi (2020)). In particular, X is exponentially distributed with rate $\frac{1}{2}$, $Z \sim \text{Pois}(x)$ for $X = x$ and Y is binomial distributed with size $n = z$ for $Z = z$ and probability $p = \frac{1}{2}$. Due to the Markov chain structure, the ground truth is $I(X;Y \mid Z) = 0$.

For each of the above experiments, we sample data according to sample size $n \in \{100, 200, \ldots, 1\,000\}$ and generate 100 data sets per sample size. We run each of the estimators on the generated data and show the mean squared error (MSE) of each estimator in Figure 4.2. Overall, our estimator performs best or very close to the best throughout the experiments and reaches an MSE lower than 0.001 with at most 1 000 samples. The best competitors are MS and MIIC; however, both are biased when we consider discrete-continuous mixture variables, as we show in Experiment V.

Experiment V

Next, we generate data according to a discrete-continuous mixture (Gao et al., 2017). Half of the data points are continuous, with X and Y being standard Gaussian with correlation $\rho = 0.8$, while the other half follows a discrete distribution with $P(1,1) = P(-1,-1) = 0.4$ and $P(1,-1) = P(-1,1) = 0.1$. In addition, we generate Z independently with $Z \sim \text{Binomial}(3, 0.2)$. Hence the ground truth is equal to $I(X;Y) = I(X;Y \mid Z) = 0.4 \cdot \log \frac{0.4}{0.5^2} + 0.1 \cdot \log \frac{0.1}{0.5^2} - \frac{1}{4}\log(1-0.8^2) \approx 0.352$.

In Figure 4.3 (top) we show the mean and MSE for this experiment. We see that our estimator starts by overestimating the true value, but its average

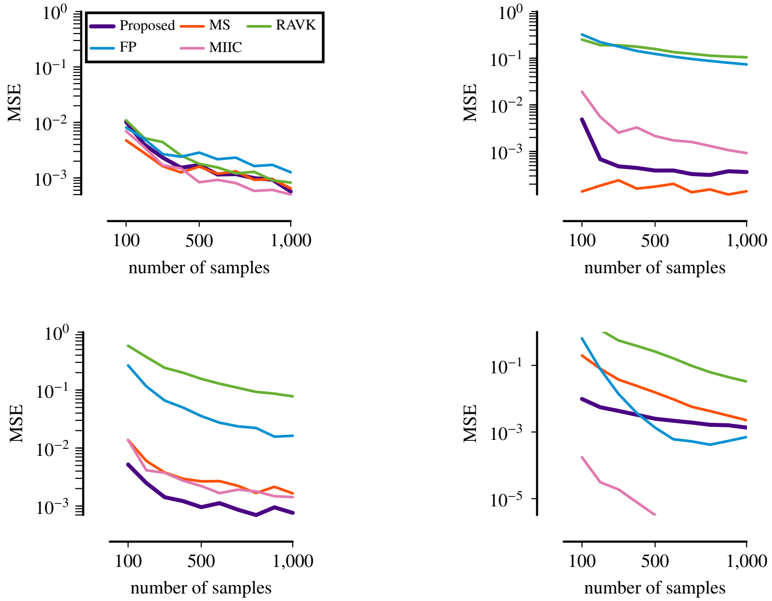


Figure 4.2: Synthetic data with known ground truth. Ordered from top-left to bottom-right, we show the MSE for Experiments I-IV, for our estimator and competing algorithms MS, RAVK, FP and MIIC.

quickly converges to the ground truth, while the competing estimators seem to have a slightly positive or negative bias. Especially FP and MIIC, which were not designed for this setup, have a clear bias even for 1 000 data points. The same trend can be observed for their MSE.

Experiment VI

Last, we test how sensitive our method is to dimensionality. We generate X and Y as in Experiment II, but fix n to 2 000 and add d independent random variables, $Z_i \sim \text{Binomial}(3,\ 0.5)$.

Figure 4.3 (bottom) shows the mean and MSE. Our estimator recovers the true CMI up to a neglectable error up to $d = 2$. After that, it starts to slowly underestimate the true CMI. This can be explained by the fact that the model costs increase linearly with the domain size and hence, we will fit fewer bins to the continuous variable for large d. We validated this conjecture by repeating the experiment for $n = 10\,000$. On this larger sample size, the MSE for our estimator remained below 0.001 even for $d = 4$. While MIIC is slightly more

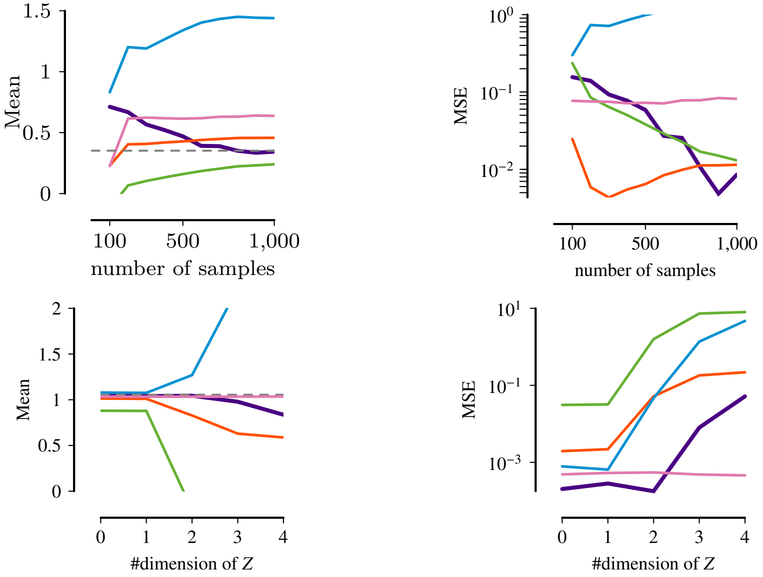


Figure 4.3: Top row: Experiment V, where we show the mean of the estimators (left) with the true CMI as a dashed gray line and the MSE (right). Bottom row: Experiment VI, where the sample size is constant at $2\,000$ and the x-axis refers to the number of dimensions of Z. We show the mean (left) and MSE (right). The color coding is chosen as in Figure 4.2.

stable for $d \geq 3$, the competing kNN-based estimators deviate quite a bit from the true estimate for higher dimensions.

Overall, we are on par with or outperform the best competitor throughout Experiments I–VI. Especially on mixture data, which is our main focus, our method is the only one that converges to the true estimate.

4.6.2 Independence Testing

Although our estimator is quite accurate, it is only close to zero and not exactly zero for the Markov chain. As we saw in the previous chapter, the empirical estimator for CMI on discrete data overestimates dependencies. Hence, we need to correct our estimator to test for independence. We do this, by considering two alternatives. First, we apply the fNML correction as suggested in Section 3.3 for SCCI_f on the discretized data and call this variant I_{SC}. As an alternative, we use an adjustment based on the Chi-squared distribution, which was originally proposed for estimating mutual information in the context of feature selection (Vinh et al., 2014). We adapted this correction for the conditional

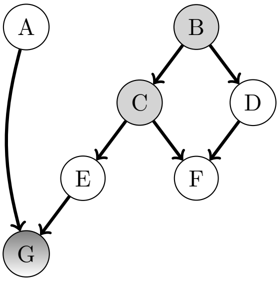


Figure 4.4: Synthetic network with continuous (white), discrete (gray) and mixed (shaded) random variables consisting of different causal structures, such as colliders, a chain ($C \rightarrow E \rightarrow G$), and a fork ($C \leftarrow B \rightarrow D$).

case, by determining the degrees of freedom as $l = (|\mathcal{X}|-1)(|\mathcal{Y}|-1)|\mathcal{Z}|$, whereas for the unconditional case, we compute $l = (|\mathcal{X}| - 1)(|\mathcal{Y}| - 1)$ as proposed by the authors. Finally, we compute

$$I_{\mathcal{X}^2}(X;Y \mid Z) = \hat{I}(X;Y \mid Z) - \frac{\mathcal{X}_{\alpha,l}}{2n} ,$$

where $\mathcal{X}_{\alpha,l}$ refers to the critical value of the Chi-squared distribution with significance level α and degrees of freedom l.

To test how well $I_{\mathcal{X}^2}$ and I_{SC} perform on mixed-type and continuous data, we benchmark both against state-of-the-art kernel-based tests RCIT and RCoT (Strobl et al., 2019), as well as JIC (Suzuki, 2016), and MIIC (Cabeli et al., 2020), which are both discretization-based methods using a correction based on stochastic complexity.[4] To apply RCIT and RCoT on mixed data, we treat the discrete data points as integers. In the following, we evaluate the performance of each test in a causal discovery setup. In addition, we provide experiments on individual collider and non-collider structures with various generating mechanisms including xor, in Appendix A.6.1.

Causal Discovery

To evaluate our test in a causal discovery setting, we generate data according to a small synthetic network—shown in Figure 4.4—that consists of a mixture of generating mechanisms that we used in experiments I-IV and includes continuous and discrete (ordinal) random variables and one mixture variable, which is partially Gaussian and partially Poisson distributed.

[4] Note that MIIC also calculates stochastic complexity based on factorized NML and JIC uses an asymptotic approximation of stochastic complexity.

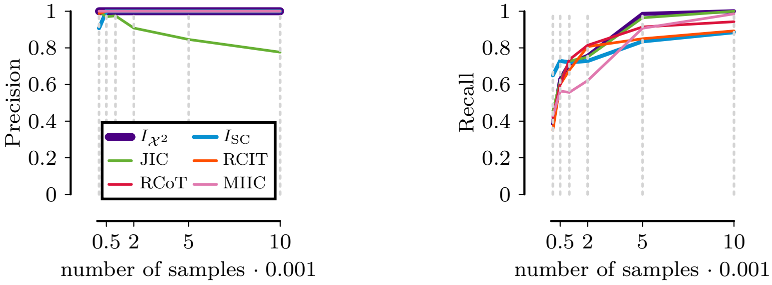


Figure 4.5: Precision (left) and recall (right) on undirected graphs inferred using the PC-stable algorithm equipped with the corresponding independence test.

More specifically, the source nodes of the network are A and B. A is generated as $A \sim \mathrm{Exp}(1)$ and $B \sim \mathrm{Unif}(0,4)$ (discrete). To get $B \to C$ we generate C as $C \sim \mathrm{Binom}(b, 0.5)$ for $B = b$, for $B \to D$ we sample D as $D \sim N(b-2, 1)$ for $B = b$ and E is sampled as exponentially distributed with rate $\frac{1}{c+1}$ for $C = c$. F is generated as a function of C and D. First, we generate C' by rounding the values of C and then we write F as $F = D^{\frac{C'}{2}} + N(0,1)$. Last, we generate G as the zero inflated Poissonization of A. Let $E' = \frac{\mathrm{sign}(E-1)+1}{2}$, which ensures that E' is either zero or one dependent on the value of E. Then $G \sim N(a,1)$ if $E' = 0$ and $A = a$, and $G \sim \mathrm{Pois}(a)$ for $A = a$ if $G = 1$.

To evaluate how well the ground truth graph can be recovered, we apply the stable PC algorithm (Colombo and Maathuis, 2014; Spirtes et al., 2000) equipped with the different independence tests, where we use $\alpha = 0.01$ for $I_{\mathcal{X}^2}$, RCIT and RCoT. Figure 4.5 shows recovery precision and recall for the undirected graph, averaged over 20 draws per sample size $n \in \{100, 500, 1\,000, 2\,000, 5\,000, 10\,000\}$.

We see that overall $I_{\mathcal{X}^2}$ performs best and being the only method that reaches both a perfect accuracy and recall. While JIC also reaches a perfect recall, it finds too many edges leading to a precision of only 80%. Although also MIIC, RCIT and RCoT have a perfect precision, their recall is worse than for $I_{\mathcal{X}^2}$. Both of the kernel-based tests also do not manage to detect all the edges even for 10 000 samples. After a closer inspection, this is due to the edge $E \to G$ that involves the discrete-continuous variable G. If we compare $I_{\mathcal{X}^2}$ to I_{SC}, we can see that the latter has a lower recall. After inspecting the data in more detail, we came to the conclusion that this is due to the fact that the fNML-based score penalizes stronger w.r.t. the domain size and thus is more conservative. The fact that we already compute the histogram model via MDL leads to relatively sparse histograms, which in addition to a strong correction leads to underfitting. Therefore, the Chi-squared correction works better in combination with our discretization algorithm.

4.7 Conclusion

We proposed a novel approach for the estimation of conditional mutual information from data that may contain discrete, continuous, and mixture variables. To be able to deal with discrete-continuous mixture variables, we defined a class of generalized adaptive histogram models. Based on our observation that CMI for mixture-variables can be written as a sum of entropies, we presented a CMI estimator based on such histograms, for which we proved that it is consistent.

Further, we used the Minimum Description Length principle to formally define optimal histograms, and proposed a greedy algorithm to practically learn good histograms from data. Finally, we demonstrated that our algorithm outperforms state-of-the-art (conditional) mutual information estimation methods, and that it can be successfully used as a conditional independence test in causal graph structure learning. Notably, for both setups, we observe that our approach performs especially well when mixture variables are present.

Part III

Bivariate Causal Discovery

In the previous parts, we focused on improving the performance of constraint-based causal discovery methods by relaxing the faithfulness assumption to allow for more complex generating mechanisms and by making advances towards better conditional independence tests. Assume that we are given the perfect independence oracle and also, faithfulness holds. Using a constraint-based causal discovery algorithm, we can still only infer the underlying causal DAG up to its Markov equivalence class. That is, if we find an edge between two nodes X and Y, which does not belong to a v-structure or can be inferred using other orientation rules based on v-structures, we cannot distinguish $X \rightarrow Y$ from $X \leftarrow Y$ and hence, need to leave this edge undirected. In this part, we will show that we can infer the edge direction between X and Y if we make assumptions about generating mechanism. The task of inferring the causal direction between two dependent random variables is also labeled as causal inference. We first discuss the general approach to do causal inference based on the algorithmic model of causality and its link to two-part MDL in Chapter 5. In Chapter 6, we propose a practical approach for causal inference on univariate numeric data based on MDL. In the following chapter, Chapter 7, we focus on identifiability, i.e., under which conditions can we guarantee that our solution is correct. Last, we extend our work to multivariate mixed-type data in Chapter 8.

Chapter 5

Causal Inference via Two-Part MDL: a Primer

The approaches that we discuss in the subsequent chapters all have in common that they are either inspired or directly build upon the algorithmic independence of conditionals (AIC) postulate. In essence, AIC postulates that from all factorizations of the joint distribution, the one with the shortest algorithmic description corresponds with the true causal DAG. As a consequence, this postulate allows for an inference scheme that goes beyond estimating the Markov equivalence class of the true DAG G. That is, in contrast to approaches that build upon conditional independence testing, we can distinguish between the two Markov equivalent DAGs $X \to Y$ and $Y \to X$. As the name suggests, the postulate is defined via Kolmogorov complexity, which we already know is not computable. Budhathoki and Vreeken (2017b) suggested to approximate AIC through two-part MDL for multivariate binary data, and this general idea has been implemented in multiple approaches, such as the ones we present in Chapters 6 and 8. Although these methods perform well in practice, the theoretical link between AIC and two-part MDL is quite crude. The former is formulated based on the true distributions, while the MDL formulation interprets this postulate from a more practical point of view, which encodes the given data with respect to a certain model and also encodes the model itself. To bridge this gap, we show that, similar to the MDL formulation, we can rewrite AIC using two-part descriptions in terms of Kolmogorov complexity. Further, we prove

This chapter mainly introduces preliminary concepts needed for Chapters 6,7 and 8. Sections 5.3 and 5.4 present new results.

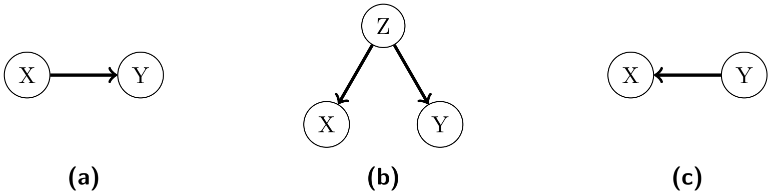


Figure 5.1: Reichenbach's Common Cause Principle: X and Y are two dependent random variables. Their dependence is either due to an unobserved confounder Z (middle), or one of the two causes the other, i.e., either X causes Y (left) or Y causes X (right).

that our new formulation leads to an equivalent inference principle as the original postulate. As a corollary, we investigate the implications of our findings for joint encodings, which encode data and model jointly. We emphasize that for such encodings is necessary to encode the model independent of the data, as otherwise, the asymmetry between the description length of the causal and anti-causal model might vanish.

This chapter is structured as follows. In Section 5.1, we lay out the general setting and explain the principle of independent mechanisms. Then, we state the AIC postulate and review how it can be approximated using two-part MDL (Section 5.2). Last, in Section 5.3, we formally validate the link between AIC and two-part MDL descriptions and we discuss practical implications of our findings for joint descriptions in Section 5.4.

5.1 The Principle of Independent Mechanisms

Throughout this chapter and the next chapters, we assume that we are given two dependent random variables X and Y for which we want to infer the underlying causal DAG from observational data. That is, we assume the data is passively collected and represents an i.i.d. sample of joint distribution P_{XY}. According to *Reichenbach's common cause principle* (Reichenbach, 1956), the dependence between X and Y can be explained by three possible graphs.

Definition 5.1 (Reichenbach's Common Cause Principle) *If two variables X and Y are statistically dependent ($X \not\perp\!\!\!\perp Y$), then there exists a third variable Z that causally influences both, that is $X \leftarrow Z \rightarrow Y$. As a special case, Z may coincide with either X, which results in the causal graph $X \rightarrow Y$ or Y ($X \leftarrow Y$). Furthermore, this variable Z d-separates X and Y. Thus, by applying the causal Markov condition, we get that $X \perp\!\!\!\perp Y \mid Z$, see Figure 5.1.*

It could also be that the dependence between X and Y is due to a combination of the above graphs, e.g. $X \rightarrow Y$, $X \leftarrow Z \rightarrow Y$. In this chapter, however, we assume all causal relations to be acyclic and further assume *causal sufficiency*, i.e., we observe all relevant variables. Thus, we only need to decide between $X \to Y$ and $X \leftarrow Y$, which is already a difficult problem since both these DAGs are Markov equivalent. Hence, it is not possible to tell apart both graphs if we rely on a constraint-based causal discovery approach (Pearl, 2009).

This is, however, not the end of the story. We can distinguish $X \to Y$ from $X \leftarrow Y$ if we additionally make assumptions about the generating mechanism. A very general such assumption, which has gained a lot of attention in recent years, is the *principle of independent mechanisms*, which focuses on the possible factorizations of the joint distribution P_{XY}. In particular, we can write P_{XY} as the product $P_X P_{Y|X}$ or $P_Y P_{X|Y}$; but why does this help? Consider an example inspired by (Peters et al., 2017, Chapter 2.1), in which the cause X corresponds to the altitude and the effect Y to the temperature as measured for different cities. Assume we consider a set of different cities from the same climate zone. If we observe the altitude for a random city, we will have a mechanism in mind to derive the corresponding temperature value (i.e. $P_{Y|X}$), which is independent of P_X. Further, we can make a thought experiment and think about how the temperature would change, if we were to change the altitude of the city, e.g., by magically lifting it into the air. Vice versa, it is hard to imagine that increasing or decreasing the temperature in a city, e.g., by putting on the heating system in every house, will change the altitude. In other words, the independence of the mechanism does not hold for the anti-causal direction: the mechanism $P_{X|Y}$ would need to take a rather particular form to be independent of P_Y, which only holds in specific settings, e.g., for a linear model with both the cause and additive noise being Gaussian distributed (Peters et al., 2017). More generally, we can formulate the principle of independent mechanisms as follows (Janzing and Schölkopf, 2010; Peters et al., 2017).

Postulate 5.1 (Principle of Independent Mechanisms) *The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other.*

In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other conditional distributions.

Projected on our two-variable example, if we assume that the principle of independent mechanisms holds, we get that $P_X \perp\!\!\!\perp P_{Y|X}$, while the same does not necessarily hold for the factorization w.r.t. the anti-causal direction. There exist several approaches that aim at using this asymmetry to infer the causal direction between two random variables from observational data (Janzing et al., 2012; Sgouritsa et al., 2015); in practice, however, it is difficult to precisely estimate this dependence. Motivated by the same idea, Janzing and Schölkopf (2010) proposed an inference principle based on algorithmic independence, which we state in the next section.

5.2 The Algorithmic Model of Causality

In the following, we will give a brief introduction to the algorithmic model of causality and the algorithmic independence of conditionals. After that, we state the commonly used two-part MDL approximation that was suggested by Budhathoki and Vreeken (2017b).

The *algorithmic model of causality* (AMC) was introduced by Janzing and Schölkopf (2010) as an algorithmic equivalent of the statistical model of causality. Simply put, we can compute the value of a node X_i with a program of constant complexity that takes as input the values of the parents of X_i and an independent noise variable.

Postulate 5.2 (Algorithmic Model of Causality) *Let G be a DAG formalizing the causal structure among the strings $x_1, \dots, x_m$. Then every x_i is computed by a program q_i with length $\mathcal{O}(1)$ from its parents pa_i and additional input n_i. We write*

$$x_i = q_i(pa_i, n_i) \ ,$$

meaning that the Turing machine computes x_i from the input pa_i, n_i using the additional program q_i and halts. The inputs n_i are jointly independent, i.e.,

$$n_i \perp\!\!\!\perp n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_m \ .$$

Janzing and Schölkopf justified this model by showing that similar to the statistical model, we can also derive an algorithmic version of the causal Markov condition. That is, the *algorithmic Markov condition* (AMC) states that the joint complexity over all nodes is given by the sum of the complexities of each individual node given the optimal compression of its parents

$$K(x_1, \dots, x_m) \overset{+}{=} \sum_{i=1}^{m} K(x_i \mid pa_i^*) \ .$$

Due to the symmetry of information, i.e., $K(x) + K(y \mid x^*) \overset{+}{=} K(y) + K(x \mid y^*)$, the algorithmic Markov condition only allows for identifying the Markov equivalence class. To be able to distinguish between Markov equivalent DAGs, Janzing and Schölkopf (2010) further postulated the algorithmic equivalent of the principle of independent mechanisms.

Postulate 5.3 (Algorithmic Independence of Conditionals) *Let G be a causal DAG over a set of m variables $\boldsymbol{X}$ with joint distribution $P_{\boldsymbol{X}}$, which is lower semi-computable, that is, $K(P_{\boldsymbol{X}}) < \infty$. A causal hypothesis is only acceptable if the shortest description of the joint distribution $P_{\boldsymbol{X}}$ is given by the* *concatenation of the shortest descriptions of the Markov kernels, i.e.,*

$$K(P_{X_1,\ldots,X_m}) \overset{+}{=} \sum_{i=1}^{m} K(P_{X_i|Pa_i}) \;,$$

where Pa_i are the parents of X_i in G. Equivalently, it holds that

$$I_A(P_{X_1|Pa_1};\ldots;P_{X_m|Pa_m}) \overset{+}{=} 0 \;.$$

If we apply the above postulate to the case where the true graph only consists of the edge $X \to Y$, we get that

$$I_A(P_X;P_{Y|X}) \overset{+}{=} 0 \;.$$

Note that it is assumed that this independence only holds for the true causal direction. For additive noise models,[1] for example, it has been shown that for the anti-causal direction we get a dependence (Janzing and Steudel, 2010), that is, $I_A(P_Y;P_{X|Y}) \gg 0$. If we combine those results, we can derive an inference rule as follows. If $X \to Y$ is the true graph, then

$$K(P_X) + K(P_{Y|X}) \overset{+}{\le} K(P_Y) + K(P_{X|Y}) \;. \tag{5.1}$$

In other words, we can infer the true causal direction by selecting the factorization with the smallest Kolmogorov complexity. To use this idea in practice, we first need to solve two problems. First, Kolmogorov complexity is not computable (Li and Vitányi, 2019), and second, we are not given the true distribution but just a limited number of data points. A principled way to solve at least the first part of the problem is to approximate Kolmogorov complexity via the MDL, as we explain below.

The first idea on how the algorithmic independence of conditionals could lead to an MDL-based inference rule was sketched out by Janzing and Schölkopf (2010), however, they do neither instantiate nor evaluate this idea in practice. They suggest that, the probabilistic models $\hat{P}_X$ and $\hat{P}_{Y|X}$, which are learned from a finite number of observations, together define a joint distribution $\hat{P}_{X\to Y}$, which is not necessarily equal to the description of $\hat{P}_{Y\to X}$ in the inverse direction. As common in MDL, they first encode the complexity of the model, i.e., $\hat{P}_X$ and $\hat{P}_{Y|X}$, and then encode the data given the model as the negative log-likelihood w.r.t. $\hat{P}_{X\to Y}$ resp. $\hat{P}_{Y\to X}$ and select the direction with the smaller complexity as the causal one.

Budhathoki and Vreeken (2017b) suggested a more practical approximation of Equation 5.1 via two-part MDL as follows. For the causal direction, we define

[1] We will discuss additive noise models in more detail in Chapter 6.

a model as $M_{X\to Y} = (M_X, M_{Y|X})$ from the class $\mathcal{M}_{X\to Y} = \mathcal{M}_X \times \mathcal{M}_{Y|X}$ that best describes the data over Y by exploiting as much structure of X as possible to save bits. By MDL, we identify the optimal model $M_{X\to Y} \in \mathcal{M}_{X\to Y}$ for data (x^n, y^n) over X and Y as the one minimizing

$$L_{X\to Y} := L(M_X) + L(x^n \mid M_X) + L(M_{Y|X}) + L(y^n \mid x^n, M_{Y|X}) \ . \tag{5.2}$$

We can define $L_{Y\to X}$ analogously and infer $X \to Y$ if $L_{X\to Y} < L_{Y\to X}$, $X \leftarrow Y$ if $L_{X\to Y} > L_{Y\to X}$, and do not decide if both terms are equal. Consequently, to use this idea in practice, we need to define the model class. Budhathoki and Vreeken (2017b) implemented their idea for multivariate binary data and used binary trees as their models. In Chapter 6, we present an approach for univariate numeric data that models the conditionals as regression functions; and in Chapter 8, we use classification and regression trees (CART) to model dependencies on multivariate mixed-type data.

Although these approaches perform well in practice, Equation 5.1 only considers the true distribution, while Equation 5.2 is formulated via a two-part description of a model and the data given this model. In the following section, we formally analyze the connection between both inference rules. We bridge the gap between both variants by deriving a two-part variant of Equation 5.1, in terms of Kolmogorov complexity, and show that on expectation both versions lead to the same inference.

5.3 Linking AIC to Two-Part Descriptions

Given an i.i.d. sample x^n w.r.t. a distribution P, the shortest encoding of the data that is theoretically possible converges to the Shannon entropy

$$H(P) = -\sum_x P(x) \log P(x) \ ,$$

as proven by Shannon's source coding theorem (Shannon, 1948). Hence, if P is a computable distribution with parameter vector θ, the sample estimate $\hat{\theta}$ will in the limit converge to the true parameter. Therefore, we could in the limit encode the data x^n conditional on P to arrive at the shortest code-length of the data given the model that describes P. Thus, the shortest encoding for our causal setup can be achieved if the model class $\mathcal{M}_X$ contains P_X and similarly, $\mathcal{M}_{Y|X}$ contains $P_{Y|X}$. Slightly abusing the notation, we define

$$L^*_{X\to Y} := L(P_X) + L(x^n \mid P_X) + L(P_{Y|X}) + L(y^n \mid x^n, P_{Y|X}) \ .$$

The above equation already comes close to an MDL version of the algorithmic independence of conditionals, however, we still need to explain how the data encoded by the model fits into the equation. To this end, we will show that the equivalent formulation of $L^*_{X\to Y}$ in terms of Kolmogorov complexity, i.e.,

$$K_{X\to Y} := K(P_X) + K(x \mid P_X) + K(P_{Y|X}) + K(y \mid x, P_{Y|X}) \; ,$$

is on expectation equal to $K(P_X)+K(P_{Y|X})+H(P_{XY})$, where $H(P_{XY})$ relates to the Shannon entropy of the joint distribution P_{XY}. The analogoue holds for the anti-causal direction, that is, on expectation $K_{Y\to X}$ is equal to $K(P_Y) + K(P_{X|Y}) + H(P_{XY})$. Thus, assuming that the algorithmic independence of conditionals holds, we get that on expectation the inequality between cause and effect holds similar to Equation 5.1. That is,

$$K_{X\to Y} \overset{+}{\leq} K_{Y\to X} \; ,$$

if $X \to Y$ is the true causal direction.

Before proving this statement, we need to state a more general Lemma that links the Kolmogorov complexity of a string x to Shannon entropy (Li and Vitányi, 2019, Chapter 8.1).[2]

Lemma 5.1 *Let $H(P)$ be the entropy of a computable probability distribution P and $H(P) < \infty$. Then,*

$$\left| \sum_x P(x)K(x \mid P) - H(P) \right| \leq \mathcal{O}(1),$$

with a constant precision that is independent of x and P.

Note that if we sum over $P(x)K(x)$ instead of $P(x)K(x \mid P)$, the inequality becomes less precise and only holds up to constant $c_P = K(P) + \mathcal{O}(1)$, which is dependent on P (Li and Vitányi, 2019, Chapter 8.1). For conditional codes such as $K(y \mid x, P_{Y|X})$ assume that given input x there exists an $\mathcal{O}(1)$ program that selects the correct probability table $P_{Y|X=x}$ from the auxiliary conditional probability table that is given as input. Based on these insights, we can derive of our main theorem.

Theorem 5.1 *Given rational distribution P_{XY} with finite support, for which all factorizations are lower semi-computable, i.e., $K(P_X)+K(P_{Y|X})+K(P_Y)+K(P_{X|Y}) < \infty$, it holds that*

$$\sum_x \sum_y P_{XY}(x,y) \left(K(P_X)+K(x \mid P_X)+K(P_{Y|X})+K(y \mid x, P_{Y|X})\right)$$

[2] The author would like to thank Bruno Bauwens for useful pointers to and clarifications w.r.t. Lemma 5.1.

is equal to $K(P_X) + K(P_{Y|X}) + H(P_{XY})$ up to an additive constant that is independent of P_X and $P_{Y|X}$. Equivalently, $K(P_Y) + K(P_{X|Y}) + H(P_{XY})$ is equal to the expectation over $K(P_Y) + K(y \mid P_Y) + K(P_{X|Y}) + K(x \mid y, P_{X|Y})$ up to an additive constant independent of P_Y and $P_{X|Y}$.

PROOF: *In the following, we prove the statement for the factorization $P_X P_{Y|X}$; the proof for $P_Y P_{X|Y}$ follows analogously. First, note that we can compute $P_X(x)$ as $P_X(x) = \sum_y P_{XY}(x, y)$. Thus, we can rewrite the first part as*

$$\begin{aligned}(\star_1) &= \sum_x \sum_y P_{XY}(x,y)\,(K(P_X) + K(x \mid P_X)) \\ &= \sum_x P_X(x)\,(K(P_X) + K(x \mid P_X)) \\ &\overset{+}{=} K(P_X) + H(P_X)\ .\end{aligned}$$

To get from line 2 to 3, we apply Lemma 5.1. Similarly, we can proceed with the second part

$$\begin{aligned}(\star_2) &= \sum_x \sum_y P_{XY}(x,y)\left(K(P_{Y|X}) + K(y \mid x, P_{Y|X})\right) \\ &= \sum_x P_X(x) \sum_y \frac{P_{XY}(x,y)}{P_X(x)}\left(K(P_{Y|X}) + K(y \mid x, P_{Y|X})\right) \\ &= K(P_{Y|X}) + \sum_x P_X(x) \sum_y \frac{P_{XY}(x,y)}{P_X(x)} K(y \mid x, P_{Y|X}) \\ &\overset{+}{=} K(P_{Y|X}) + \sum_x P_X(x) \sum_y P_{Y|X=x}(y) K(y \mid P_{Y|X=x}) \\ &\overset{+}{=} K(P_{Y|X}) + \sum_x P_X(x) H(P_{Y|X=x}) \\ &= K(P_{Y|X}) + H(P_{Y|X})\ .\end{aligned}$$

Importantly, in the step from line 3 to 4, we assume that we can select the correct probability table $P_{Y|X=x}$ form inputs $P_{Y|X}$ and x with an $\mathcal{O}(1)$ program.

If we combine $(\star_1)$ and $(\star_2)$, we get $K(P_X) + K(P_{Y|X}) + H(P_{XY})$ and obtain an equivalent result for the inverse direction due to the symmetry of the joint entropy. □

Although the theorem is only stated for two random variables, it is straightforward to extend it to the general formulation of the algorithmic independence of conditionals. In particular, we have

$$\sum_{i=1}^{m}\sum_{x_i}\sum_{pa_i} P_{X_i\mathrm{Pa}_i}(x_i, pa_i)\left(K(P_{X_i|\mathrm{Pa}_i}) + K(x_i \mid pa_i, P_{X_i|\mathrm{Pa}_i})\right)$$

is equal to $\sum_{i=1}^{m} K(P_{X_i|\mathrm{Pa}_i}) + H(P_{X_1,\ldots,X_m})$ up to a constant.

In the following section, we point out that the results of Theorem 5.1 do not necessarily hold for joint descriptions and discuss implications of these observations for practical MDL encodings.

5.4 Connection to Joint Descriptions

The optimization goal of a two-part encoding, e.g. two-part MDL, is also often formalized as finding that model $M^* \in \mathcal{M}$, which minimizes the *joint* costs of data and model, that is,

$$M^* = \operatorname*{argmin}_{M \in \mathcal{M}} L(D, M) = L(M) + L(D \mid M)\ .$$

Hence, intuitively we could rewrite $L_{X\to Y}$ as $L(x^n, M_X) + L(y^n, M_{Y|X} \mid x^n)$. The problem is, if we rigorously expand the second term, we need to encode the model given the data, i.e.,

$$L(y^n, M_{Y|X} \mid x^n) = L(M_{Y|X} \mid x^n) + L(y^n \mid x^n, M_{Y|X})\ .$$

In terms of MDL, we can argue that the model is independent of the data x^n. In general, while technically possible, it is not common to encode a model conditioned on the data. Thus, we do not elaborate further on this ambiguity and jump into Kolmogorov land.

In particular, assume that $X \to Y$ is the true causal model. If we were to prove that $K(x, P_X) + K(y, P_{Y|X} \mid x)$ is on average equal to $K(P_X) + K(P_{Y|X}) + H(P_{XY})$, the proof would become slightly more involved. It is inevitable, however, that to split off $P_{Y|X}$ from $K(y, P_{Y|X} \mid x)$ we need to keep x in the conditional term. That is, we arrive at the term $K(P_{Y|X} \mid x)$ and would need to argue that it is equal to $K(P_{Y|X})$. Since $P_X \perp\!\!\!\perp P_{Y|X}$ and x is sampled from P_X, we can indeed conclude that $K(P_{Y|X} \mid x) = K(P_{Y|X}) + \mathcal{O}(1)$. For the anti-causal direction, this independence does not hold, i.e., $P_Y \not\!\perp\!\!\!\perp P_{X|Y}$.

Hence, $K(P_{X|Y} \mid y) \overset{+}{\leq} K(P_{X|Y})$. Due to this asymmetry, we get that

$$\begin{aligned}
&\sum_x \sum_y P_{XY}(x,y)(K(y,P_Y) + K(x,P_{X|Y} \mid y)) \\
&\overset{+}{\leq} K(P_Y) + K(P_{X|Y}) + H(P_{XY}) \\
&\overset{+}{\geq} K(P_X) + K(P_{Y|X}) + H(P_{XY}) \; .
\end{aligned}$$

In other words, an approximation of this formulation does not allow us to distinguish $X \to Y$ from $Y \to X$, because we cannot guarantee that the inequality between the causal and anti-causal direction still holds. Thus, encodings that approximate the joint description with the goal to do causal inference should be designed with caution, as the description length of the conditional model should be independent of the data of the conditioning variable.

5.5 Conclusion

To sum up, we first established a clear connection between $K_{X \to Y}$ and $L^*_{X \to Y}$, which is a possible instantiation of the two-part MDL description for the causal model $L_{X \to Y}$. In addition, we showed that comparing $K_{X \to Y}$ to $K_{Y \to X}$ is equivalent to comparing the Kolmogorov complexities of the corresponding factorizations, i.e., the inference rule based on AIC. Hence, under reasonable model assumptions, we can approximate AIC with two-part MDL according to $L_{X \to Y}$ and $L_{Y \to X}$. In the following Chapters, we will present different approaches that are inspired or base upon the algorithmic model of causality. However, note that the research presented in those chapters was done before the work presented in this chapter was developed.

Chapter 6

Causal Inference via MDL-based Regression

We now turn from theory to practice and consider the problem of inferring the most likely causal direction between two statistically dependent univariate numeric random variables X and Y, given only an i.i.d. sample from their joint distribution. We further assume acyclicity and causal sufficiency, that is, we assume that there is no hidden confounder Z that causes both X and Y. Simply put, we are interested in identifying whether X causes Y ($X \to Y$), or whether Y causes X ($Y \to X$).

In the previous parts, we discussed faithfulness and conditional independence tests, which are both relevant for constraint-based causal discovery methods. However, even if faithfulness holds and we are given the perfect independence criterium, we cannot decide between the two Markov equivalent DAGs $X \to Y$ and $Y \to X$ (Pearl, 2009) since these graphs do not contain a v-structure. One possibility to infer cause and effect in such a scenario is by doing interventions (Pearl, 2009). The idea here is that we would actively change the distribution or intervene on the distribution of X or Y and check if this changes the distribution of the second variable. If not, we know that we performed the intervention on the effect of both variables. In the setup we consider, however, we do not have access to such interventional data. Thus, we need to make assumptions on the generating mechanism to be able to find an asymmetry between cause and effect.

This chapter is based on Marx and Vreeken (2017, 2019b).

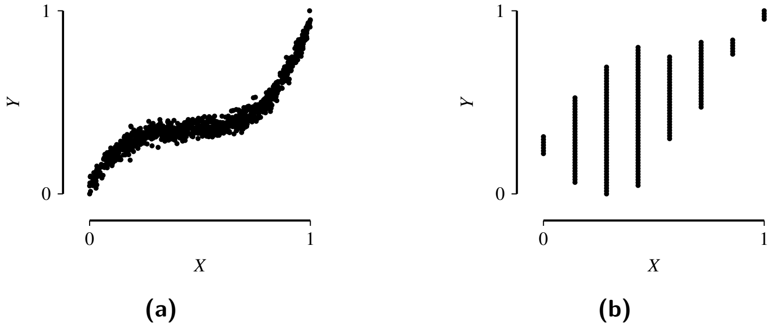


Figure 6.1: Example data for the ground truth model $X \rightarrow Y$. The left-hand data is generated using a cubic function with Gaussian noise, whereas the right-hand data is generated using a global trend function as well as an additional local function to model the structure within duplicated values of X.

One such assumption is the algorithmic independence of conditionals (Janzing and Schölkopf, 2010) that we discussed in the previous chapter. That is, we assume that the algorithmic mutual information between the distribution of the cause and its mechanism ($P_{Y|X}$) is zero—i.e., they are algorithmically independent. As a consequence, the shortest algorithmic description of the joint distribution of X and Y is achieved by first describing the distribution of the cause and then describing the conditional distribution of the effect given the cause. Building upon this postulate, Budhathoki and Vreeken (2017b) proposed to estimate this inequality using two-part MDL and instantiated their framework for multivariate binary data. In this work, we build upon this idea and develop a two-part MDL approximation for numeric data.

Simply put, we propose to fit a regression model from X to Y, and vice versa, measuring both the complexity of the function, as well as the error it makes in bits, and select that direction which has the shortest description length. We carefully construct an MDL score such that we can meaningfully compare between different types of functional dependencies, including linear, quadratic, cubic, reciprocal, and exponential functions, and the error that they make. This way, for example, we will find that we can more succinctly describe the data in Figure 6.1(a) by a cubic function than with a linear function. Although it takes fewer bits to describe the linear function, it will take many more bits to describe the larger error it makes. However, not all data might look as smooth as in the previous example. In fact, many real-world data sets look like the data plotted in Figure 6.1(b), see Mooij et al. (2016). We can clearly see that despite the general linear trend, the data contains more structure, which we can encode. That is, for all values x of X, the associated values of Y show a similar pattern. In contrast, if we rotate the plot by 90 degrees, we do not observe such regularities for the X values mapped to a single Y value. We can exploit this asymmetry by considering local regression functions per value of X, each individually fitted but since we assume all noise to be of the same type, all should be in the same function class. In this particular example, we therewith correctly infer that X causes Y. The Minimum Description Length principle prevents us from overfitting, as such local functions are only included if they aid global compression.

The remainder of this chapter is structured as follows. In Section 6.1, we introduce our score based on the algorithmic independence of conditional, as well as a practical instantiation based on the MDL principle. Since only providing an inference result does not give us any information about how certain we are that this decision is correct, we provide a detailed discussion on the confidence of a provided inference in Section 6.2. Additionally, we discuss and provide two different significance tests which we evaluate in our experiments. After that, we introduce the linear-time algorithms Slope and Sloper to efficiently compute our score in Section 6.3. In Section 6.4, we discuss related work and in Section 6.5, we empirically validate our approach on synthetic data as well as on real-world benchmark data. Especially on the benchmark data, we clearly outperform the state-of-the-art in the field.

6.1 Information Theoretic Causal Inference

In this chapter, we assume that we are given two correlated univariate numeric random variables X and Y. Further, we assume that causal sufficiency holds and no causal relation is cyclic. Hence, according to Reichenbach's common cause principle (see Definition 5.1), the true causal graph between X and Y can either be $X \rightarrow Y$ or $X \leftarrow Y$.

To estimate which of the two options corresponds with the true graph, we build upon the algorithmic independence of conditionals (see Section 5.2). Accordingly, if $X \rightarrow Y$ is the true graph, we get that

$$K(P_X) + K(P_{Y|X}) \overset{+}{\leq} K(P_Y) + K(P_{X|Y}) \; . \tag{6.1}$$

This means that if $X \rightarrow Y$, first describing the marginal distribution of the cause $K(P_X)$ and then describing the conditional distribution of the effect given the cause $K(P_{Y|X})$, will be shorter than the Kolmogorov complexity of the factorization for the anti-causal direction.

Although Equation (6.1) already allows for inferring the causal direction for a given pair, we obtain a more robust score, allowing for fair comparison of results independent of data sizes, when we normalise the result. In particular, Budhathoki and Vreeken (2017b) recently proposed to normalize the scores with the sum of the description lengths for the marginal distributions. We therefore define our causal indicator as

$$\mathcal{I}^K_{X\to Y} = \frac{K(P_X) + K(P_{Y|X})}{K(P_X) + K(P_Y)} \ ,$$

and $\mathcal{I}^K_{Y\to X}$ in the same manner. Consequently, we infer $X \to Y$, if $\mathcal{I}^K_{X\to Y} < \mathcal{I}^K_{Y\to X}$, and $Y \to X$, if $\mathcal{I}^K_{X\to Y} > \mathcal{I}^K_{Y\to X}$ and do not decide if $\mathcal{I}^K_{X\to Y} = \mathcal{I}^K_{Y\to X}$.

The confidence of our score is $\mathcal{C} = |\mathcal{I}^K_{X\to Y} - \mathcal{I}^K_{Y\to X}|$. The higher, the more certain we are that the inferred causal direction is correct. In Section 6.2, we will show how we can in addition define two analytical tests to determine whether an inference is statistically significant.

Next, we will show how we can instantiate the above indicator in practice using a two-part MDL encoding.

6.1.1 Causal Inference by MDL

As Kolmogorov complexity is not computable, we will instantiate $\mathcal{I}^K_{X\to Y}$ and $\mathcal{I}^K_{Y\to X}$ using the Minimum Description Length principle (Li and Vitányi, 2019; Grünwald, 2007). In practice this means we will need to define a model class $\mathcal{M}$ under which we can estimate $\mathcal{I}^K_{X\to Y}$ as

$$\mathcal{I}^L_{X\to Y} = \frac{L(X) + L(Y \mid X)}{L(X) + L(Y)} \ .$$

As suggested by Budhathoki and Vreeken (2017b), we describe the model for the causal direction $M_{X\to Y} \in \mathcal{M}$ in two parts. We use M_X to describe the marginal and $M_{Y|X}$ to describe the conditional. Thus, we define $L(X)$ as the sum $L(M_X) + L(x^n \mid M_X)$ and further define $L(Y \mid X)$ as $L(M_{Y|X}) + L(y^n \mid x^n, M_{Y|X})$, where (x^n, y^n) corresponds to an empirical sample of n entries that was drawn i.i.d. from P_{XY}.[1]

We define $\mathcal{I}^L_{Y\to X}$ analogue to $\mathcal{I}^L_{X\to Y}$, and we infer $X \to Y$, if $\mathcal{I}^L_{X\to Y} < \mathcal{I}^L_{Y\to X}$, $Y \to X$, if $\mathcal{I}^L_{X\to Y} > \mathcal{I}^L_{Y\to X}$ and do not decide if $\mathcal{I}^L_{X\to Y} = \mathcal{I}^L_{Y\to X}$ or if the difference is below a user-defined threshold. Like above, confidence $\mathcal{C}$ is simply the absolute difference between $\mathcal{I}^L_{X\to Y}$ and $\mathcal{I}^L_{Y\to X}$.

To put this framework into practice, we need to define the model class and explain how we encode the data given a corresponding model.

[1] As described in Section 5.3, this two-part MDL description can be justified via rewriting the algorithmic independence of conditionals as two-part descriptions in terms of Kolmogorov complexity.

Intuition for the Conditional Encoding

The general idea is simple: we use regression to model the data of Y given X. That is, we model Y as a function f of X and independent noise N, i.e., $Y = f(X) + N$. We do so by fitting a regression function f over X and treat the error as Gaussian distributed noise. Naturally, the better $f(X)$ fits Y, the fewer bits we will have to spend on encoding the errors. The more parameters $f(X)$ has, however, the more bits we will have to spend on encoding these. This way, MDL naturally balances the complexity of the model to that of the data given the model (Grünwald, 2007). For example, while a linear function is simpler to describe than a cubic function, the latter will be a significantly better fit for the data shown in Figure 6.1(a), and hence, MDL will prefer the more complex model. At the same time, we would not decide to fit a polynomial of a higher degree, since the model costs increase, while the error will at most reduce by a small margin.

A key idea in our approach is to compress the data as much as possible. Thus, we do not only consider a single global regression function f_g, but also consider fitting additional local, or compound functions as models. That is, we consider models that besides the global function f_g may additionally consist of *local* regression functions f_l that model Y for those values x of X that non-deterministically map to multiple values of Y. That is, per such value of X, we take the corresponding values of Y, sort these ascendingly, and uniformly re-distribute them on X over a fixed interval. For these re-distributed points, we can fit a local regression model f_l, if it improves the overall compression. This way, we will for example be able to much more succinctly describe the data in Figure 6.1(b) than with a single global function. In particular, we can pick up local structures, e.g., that all values of Y associated to a single point in X are roughly uniformly distributed. To avoid overfitting we use MDL, and only allow a local function for a value of X into our model if it provides a gain in overall compression. Further, we assume that for the true causal model the data in the local components follows the same pattern. Hence, we only allow models in which all local functions are of the same type, e.g., all are linear.

In the following paragraphs, we specify how we describe a model and vice versa, how we encode the data given the model.

Encoding the Marginals

We start by defining the cost for the marginal distributions, $L(X)$ and $L(Y)$, which mostly serve to normalize our causal indicators $\mathcal{I}^L_{X\rightarrow Y}$ and $\mathcal{I}^L_{Y\rightarrow X}$. As we beforehand do not know how X or Y are distributed, and do not want to incur any undue bias, we encode both using a uniform prior with regard to the data resolution τ of X and Y. That is, we have $L(x^n \mid M_X) = n \log \frac{\max(X)-\min(X)}{\tau_X}$, where τ_X is the resolution of the data of X. Since we normalize the data before running our algorithm (see Section 6.3), $\max(X) = 1$, $\min(X) = 0$ and thus $L(x^n \mid M_X) = -n \log \tau_X$. Note that resolution τ can be different between X and Y—we specify how we choose τ in the next section. Since we assume the same model for M_X and M_Y, the model costs will cancel overall, hence there is no need to specify them. We define $L(y^n \mid M_Y)$ in the same way as the marginal encoding for the data over X.

Encoding the Conditional Model

Formally, we write F for the set of regression functions, or model, we use to encode the data of Y given X. A model F consists of at least one global regression function $f_g \in \mathcal{F}$, and up to k local regression functions $f_l \in \mathcal{F}$, where k refers to the number of unique values of x^n. We write F_l for the set of local regression functions $f_l \in F_l$, and require that all $f_l \in F_l$ are of the same type. The description length, or encoded size, of F is

$$L(F) = L_{\mathbb{N}}(|F_l|) + \log \binom{k}{|F_l|} + 2\log(|\mathcal{F}|) + L(f_g) + \sum_{f_l \in F_l} L(f_l) \ ,$$

where we first describe the number of local functions using $L_{\mathbb{N}}$,[2] the MDL optimal encoding for integers $z \geq 1$ (Rissanen, 1983), then map each f_l to its associated value of X, after which we use $\log |\mathcal{F}|$ bits to identify the type of the global regression function f_g, and whenever F_l is non-empty also $\log |\mathcal{F}|$ bits to identify the type of the local regression functions f_l. Finally, we encode the functions themselves. Knowing the type of a function, we only need to encode its parameters, and hence

$$L(f) = \sum_{\phi \in \Phi_f} L_{\mathbb{N}}(s) + L_{\mathbb{N}}(\lceil |\phi| \cdot 10^s \rceil) + 1 \ ,$$

where we encode each parameter ϕ up to a certain precision p, where $p > 0$ is an integer. We shift ϕ by the smallest integer number s such that $\phi \cdot 10^s \geq 10^p$, i.e., $p = 3$ means that we consider four digits. Accordingly, we encode the shift, the shifted digit and the sign.

Encoding the Residuals

Reconstructing the data of Y given $f(X)$ corresponds to encoding the residuals, or the error the model makes. Since we fit our regression functions by

[2] $L_{\mathbb{N}}$ of an integer $z > 0$ is defined as $L_{\mathbb{N}}(z) = \log^* z + \log c_0$, where $\log^* z = \log z + \log\log z + \ldots$ and we consider only the positive terms, and c_0 is a normalization constant to ensure the Kraft inequality holds (Kraft, 1949).

minimizing the sum of squared errors, which corresponds to maximizing the likelihood under a Gaussian, it is a natural choice to encode the errors using a Gaussian distribution with zero-mean.

Since we have no assumption on the standard deviation of the error, we set the variance in the encoding for a normal distribution (Grünwald, 2007, Chapter 12) to be the empirical estimate $\hat{\sigma}$. That is, for data points x^n with large n we get that

$$\begin{aligned} -\ln(x^n, 0, \hat{\sigma}^2) &= \frac{SSE}{2\hat{\sigma}^2} + \frac{n}{2}\ln 2\pi\hat{\sigma}^2 \\ \Leftrightarrow -\ln(x^n, 0, \hat{\sigma}^2) &= \frac{n}{2} + \frac{n}{2}\ln 2\pi\hat{\sigma}^2 \\ \Leftrightarrow -\log(x^n, 0, \hat{\sigma}^2) &= \frac{n}{2\ln 2} + \frac{n}{2}\log 2\pi\hat{\sigma}^2 = \frac{n}{2}\left(\frac{1}{\ln 2} + \log 2\pi\hat{\sigma}^2\right) \ , \end{aligned}$$

where we do a basis change to the logarithm with basis two. Hence, the encoded size of the error of $F(X)$ with respect to the data of Y corresponds to

$$L(Y \mid F, X) = \sum_{f \in F} \left(\frac{n_f}{2} \left(\frac{1}{\ln 2} + \log 2\pi\hat{\sigma}^2 \right) - n_f \log \tau \right) \ ,$$

where n_f is the number of data points for which we use a specific function $f \in F$ and the term with regard to τ is a correction term with respect to the resolution of the data. Intuitively, this score is higher the less structure of the data is described by the model and increases logarithmically to the sum of squared errors.[3]

Combining the data and model costs, we can now proceed and define the total encoded size of the conditional encoding of Y given X as

$$L(Y \mid X) = L(F) + L(Y \mid F, X) \ . \tag{6.2}$$

By MDL we are after that model F that minimizes Equation (6.2). Next, we discuss how we can assess the significance of a decision using the no-hypercompression inequality.

[3] In retrospect, we could have also assumed that the error has standard deviation one and hence encode it as $\sum_{f \in F} \frac{SSE}{2\ln 2} + \frac{n_f}{2}\log 2\pi$ (Grünwald, 2007, Chapter 12), which is numerically more stable, since we do not run the risk of taking the logarithm of a number < 1.

6.2 Significance of a Prediction

In the following, we discuss two possibilities to evaluate a decision. Since there are only two options, X is the cause, or Y is the cause; and very rarely, the scores are equal, we want to provide the user some insight to the decision. One possibility is to compute an MDL-based p-value and the second option is by looking at the relative confidence.

6.2.1 Significance by Hypercompression

Ranking based on confidence works well in practice. Ideally, we would additionally like to know the significance of an inference. It turns out we can define an appropriate hypothesis test using the no-hypercompressibility inequality (Bloem and de Rooij, 2020; Grünwald, 2007). In a nutshell, under the hypothesis that the data was sampled from the null-model, the probability that any other model can compress k bits better is

$$P_0(L_0(x) - L(x) \geq k) \leq 2^{-k} \ .$$

This means that if we assume the null model to be the direction corresponding to the least-well compressed causal direction, we can evaluate the probability of gaining k bits by instead using the most-well compressed direction. Formally, if we write $L_{X \to Y}$ for $L(X) + L(Y \mid X)$, and vice-versa for $L_{Y \to X}$, we have

$$L_0 = \max\{L_{X \to Y}, L_{Y \to X}\} \ .$$

The probability that the data can be compressed

$$k = |L_{X \to Y} - L_{Y \to X}|$$

bits better by encoding it in the anti-causal direction is then simply 2^{-k}.

In fact, we can construct a more conservative test by assuming that the data is not causated, but merely correlated. That is, we assume *both* directions are wrong; the one compresses too well, the other compresses too poorly. Hence, if we assume these two to be equal in terms of exceptionality, the null complexity is the mean between the complexities of the two causal directions, i.e.,

$$L_0 = \min\{L_{X \to Y}, L_{Y \to X}\} + \frac{|L_{X \to Y} - L_{Y \to X}|}{2} \ .$$

The probability of the best-compressing direction is then 2^{-k} with

$$k = \frac{|L_{X \to Y} - L_{Y \to X}|}{2} \ .$$

We can now set a significance threshold α as usual, such as $\alpha = 0.001$, and use this to prune out those cases where the difference in compression between the two causal directions is insignificant.

6.2.2 Significance by Confidence

Although the above significance test based on the absolute difference in compression follows nicely from theory, and behaves well in practice, it is not free of problems. In particular, as most significance tests, it is sensitive to the number of samples, which in our context can be directly linked to the initial complexities $L(X)$ and $L(Y)$. Assume we draw two samples from the distribution P_{XY}, where sample A only consists of 1 000 data points, while sample B contains $n = 10\,000$ data points. Since both, the encoding of the marginal, as well as the encoding for the conditional, depend on n, it is likely that the absolute difference between $L_{X\to Y}$ and $L_{Y\to X}$ is larger for sample B. However, if we find an equally large absolute difference between $L_{X\to Y}$ and $L_{Y\to X}$ for a pair of nodes on a relatively small sample and a different pair on a large sample, we would intuitively say that the large gain on the small sample is more significant. Following this conjecture, the above significance test using the absolute difference is biased towards larger data sets.

To resolve this bias we formulate a null hypothesis with respect to the marginal complexities. In particular, we rescale the initial complexity $L(X) + L(Y)$ as if they would sum up to b bits. Thus, we write the new null hypothesis as H_0*: Given a budget of b bits both directions compress equally well.* With this hypothesis, we calculate k as

$$k = \frac{|L_{X\to Y} - L_{Y\to X}|}{2} \cdot \frac{b}{L(X) + L(Y)} = \frac{\mathcal{C} \cdot b}{2} \; .$$

This means that finding a threshold for the confidence value is equivalent to the relative significance test. In particular, we can calculate a confidence threshold given a significance level α and a budget b as $\mathcal{C} = -2\log(\alpha)/b$. For instance, allowing a budget of $b = 1\,000$ bits and a significance level of $\alpha = 0.05$ renders all inferences with a confidence value lower than 0.00864 insignificant. Informally, we say that we do not expect that a difference of more than k in b bits, is due to a random effect. In other words, to be able to compare significance values between different experiments, we pretend that both experiments contained the same amount of samples.

We will evaluate both of the above procedures, in addition to our confidence score, in the experiments. In the next section we describe how we compute the marginal and conditional costs in linear time.

6.3 The Slope Algorithm

With the framework defined in the previous section, we can determine the most likely causal direction and the corresponding confidence value. In this section, we present the Slope algorithm to efficiently compute the causal indicators. To keep the computational complexity of the algorithm linear, we restrict ourselves to linear, quadratic, cubic and exponential functions; and their counterparts: reciprocal and logarithmic functions. Note that at the cost of extra computation this class may be expanded arbitrarily. We start by introducing the subroutine of Slope that computes the conditional complexity of Y given X.

6.3.1 Calculating Conditional Scores

Algorithm 6.1 describes the subroutine to calculate the conditional costs $L(Y \mid X)$ or $L(X \mid Y)$. We start by fitting a global function f_g for each function class $c \in \mathcal{F}$ and choose that f_g with the minimum sum of data and model costs (line 2). Next, we add f_g to the model F and store the total costs (3–4). For purely continuous data, we are done.

If X includes duplicate values, however, we need to check whether fitting local models leads to a gain in compression. To this end we check for each value x_i that occurs at least twice in the sample x^n if we can fit a local function. In particular, we fit the corresponding values in y^n, ordered ascendingly, as a function $f_l(X_i)$, where X_i corresponds to a sequence of ascending data points, which are uniformly spaced over the interval $[-t, t]$, where t is a user-determined scale parameter (lines 9–13). If the model costs of the new local function f_l are higher than the gain on the data side, we do not add f_l to our model (line 14). As it is fair to assume that the data generating process is the same for each local component, we restrict all local functions to be of the same type. As final result, we return the costs according to the model with the smallest total encoded size. In case the process does not involve a local function, our model will only contain f_g.

6.3.2 Causal Direction and Confidence

In the previous paragraph, we described Algorithm 6.1, which is the main algorithmic part of Slope. Before applying it, we first normalize X and Y to be from the same domain and then determine the data resolutions τ_X and τ_Y for X and Y. To obtain the data resolution, we calculate the smallest non-zero difference between two instances of the corresponding random variable. Next, we apply Algorithm 6.1 for both directions to obtain $L(Y \mid X)$ and $L(X \mid Y)$. Subsequently, we estimate the marginals $L(X)$ and $L(Y)$ based on their data resolutions. This we do by modelling both with a uniform prior

Algorithm 6.1: ConditionalCosts(Y, X)

```
input  : data over random variables $Y$ and $X$
output: score $L(Y \mid X)$
$F$ = empty model;
$f_g = \text{FitFunction}(Y \sim X, \mathcal{F})$;
$F = F \cup f_g$;
$s = s_g = L(F) + L(Y \mid F, X)$;
$X_u = \{x \mid \text{count } x \in x^n \geq 2\}$;
foreach $\mathcal{F}_c \in \mathcal{F}$ do
    $s_c = s_g,\ F_c = F$;
    foreach $x_i \in X_u$ do
        $Y_i = \{y \in Y \mid y \text{ maps to } x_i\}$;
        sort $Y_i$ ascendingly;
        $X_i = \text{norm}(1 : |Y_i|, \min = -t, \max = t)$;
        $f_l = \text{FitFunction}(Y_i \sim X_i, \mathcal{F}_c)$;
        $\hat{s} = L(F_c \cup f_l) + L(Y \mid F_c \cup f_l, X)$;
        if $\hat{s} < s_c$ then $s_c = \hat{s},\ F_c = F_c \cup f_l$;
    if $s_c < s$ then $s = s_c$;
return $s$;
```

as $L(X) = -n \log \tau_X$ and $L(Y) = -n \log \tau_Y$. In the last step, we compute $\mathcal{I}^L_{X\to Y}$ and $\mathcal{I}^L_{Y\to X}$ and report the causal direction as well as the corresponding confidence value $\mathcal{C}$.

The choice of the resolution might seem to be ad-hoc, which it is. However, since we compute the unconditioned complexities with a uniform prior, the exact value of the resolution is not important. In general, setting a resolution in our score prevents us from getting negative code lengths in case $\hat{\sigma}$ approaches zero. In this special setting, where we only consider two univariate variables with the same sample size, the penalty for the resolution cancels out. In particular, in both $\mathcal{I}^L_{X\to Y}$ and $\mathcal{I}^L_{Y\to X}$, we subtract n times the negative logarithm of the resolution for X and Y. Hence, the number of bits spent to correct for the resolution is equal for both $\mathcal{I}^L_{X\to Y}$ and $\mathcal{I}^L_{Y\to X}$.

6.3.3 Combining Basis Functions

To extend the generality of Slope, we provide a second version of it, which we call Sloper. The aim of Sloper is to allow for more complex functions, e.g.

$Y = a + bX + c\log(X) + dx^{-3} + N$. This we do by fitting a mixture of basis functions as the global function. As a consequence, Sloper is more flexible and can help to infer more complex functional relationships. Naturally, this comes at a cost. In particular, we go over each possible combination of basis functions—in our case $2^{|\mathcal{F}|} - 1$ with $|\mathcal{F}| = 8$ basis functions—and find the one minimizing the two part-costs.

Since all possible combinations can be non-ambiguously enumerated, we can still use the same encoding.

6.3.4 Computational Complexity

To assess the computational complexity, we have to consider the score calculation and the fitting of the functional relations. The model costs are computed in linear time according to the number of parameters, whereas the data costs need linear time with regard to the number of data points n. Since we restrict ourselves to relatively simple functions, we can fit these in linear time w.r.t. n. To fit local functions, in the worst case we perform $n/2$ times $|\mathcal{F}|$ fits over two data points, which is still linear. In total, the runtime complexity of Slope hence is $\mathcal{O}(n|\mathcal{F}|)$, for Sloper respectively $\mathcal{O}(n2^{|\mathcal{F}|})$. In practice, Slope and Sloper are very fast and typically output a result within a few seconds, up to a few minutes for pairs with tens of thousands of samples.

6.4 Related Work

A well studied framework to infer the causal direction between two correlated random variables are *additive noise models* (ANMs) (Shimizu et al., 2006). That is, we assume that Y was generated as a function of X with additive noise N_X independent of X, i.e., $Y = f(X) + N_X$ with $X \perp\!\!\!\perp N_X$. It turns out that for various settings (Peters et al., 2011b, 2014) the correct causal direction is identifiable as there does not exist an ANM in the anti-causal direction; it is impossible to find a function $X = g(Y) + N_Y$ where $Y \perp\!\!\!\perp N_Y$ holds. This is the case for linear functions f and non-Gaussian noise N_X (Shimizu et al., 2006), non-linear functions and additive noise (Hoyer et al., 2009), post-non-linear models where the effect is generated as $Y = f_1(f_2(X) + N_X)$ (Zhang and Hyvärinen, 2009), mixtures of multiple additive noise models (Hu et al., 2018), as well as for discrete regression (Peters et al., 2011a). In a comparative study, Mooij et al. (2016) reviewed several ANM-based approaches for continuous data from which ANM-pHSIC performed best. To test for independence of cause and noise, ANM-pHSIC employs a kernel-based independence criterium, the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2008), which can find complex non-linear dependencies.

A limiting factor of these approaches is that the results strongly rely on the used independence test and the fitting algorithm (Mooij et al., 2016). Problems can arise when the functions overfit and an ANM is discovered for both directions, as we show in our experiments. In addition, it is hard to derive a meaningful confidence score from the corresponding p-values, as they are highly dependent on the sample size (Anderson et al., 2000).

Another class of methods relies on the postulate of the independence of mechanisms, that we discussed in Section 5.1. Quickly summarized, this postulate assumes that if $X \rightarrow Y$, the distribution of the cause (P_X) is independent of the mechanism ($P_{Y|X}$), while the same does not hold for the anti-causal direction (Janzing and Schölkopf, 2010). The authors of IGCI define this independence via orthogonality in the information space. Practically, they define their score using the entropies of X and Y (Janzing et al., 2012). Liu and Chan (2016) implemented this framework by calculating the distance correlation for discrete data between P_X and $P_{Y|X}$. A third approach based on this postulate is Cure (Sgouritsa et al., 2015). Here, the main idea is to estimate the conditional using unsupervised inverse Gaussian process regression on the corresponding marginal and compare the result to the supervised estimation. If the supervised and unsupervised estimation for $P_{X|Y}$ deviate less than those for $P_{Y|X}$, an independence of $P_{Y|X}$ and P_X is assumed and the causal direction $X \rightarrow Y$ is inferred. Although well formulated in theory, the proposed framework is only solvable for data consisting of at most 200 data points and otherwise strongly relies on finding a good sample of the data.

We base our approach on a related postulate, that is, the algorithmic independence of conditionals (Janzing and Schölkopf, 2010; Lemeire and Dirkx, 2006). The postulate states that if $X \rightarrow Y$, the complexity of the description of the joint distribution in terms of Kolmogorov complexity, $K(P_{XY})$, will be shorter when first describing the distribution of the cause $K(P_X)$ and than describing the distribution of the effect given the cause $K(P_{Y|X})$ than vice versa (see also Section 5.2). To the best of our knowledge, Mooij et al. (2010) were the first to propose a practical instantiation of this framework based on the Minimum Message Length principle (MML) (Wallace and Boulton, 1968) using Bayesian priors. Vreeken (2015) proposed to approximate the Kolmogorov complexity for numeric data using the cumulative residual entropy, and gave an instantiation for multivariate continuous-valued data. Perhaps most related to Slope is Origo (Budhathoki and Vreeken, 2017b), which uses two-part MDL to infer the causal direction on binary data, whereas we focus on univariate numeric data. Last, a quite recent proposal, QCCD (Tagasovska et al., 2020), approximates AIC using non-parametric conditional quantile estimation. Since this approach was published after the paper in which we proposed Slope, we will not compare to QCCD in this Chapter, but we will include comparisons to QCCD in the next Chapter.

6.5 Experiments

In this section, we empirically evaluate Slope and Sloper. In particular, we consider synthetic data, a benchmark data set, and a real-world case study. We implemented both in *R* and make the code, the data generators, as well as the real world data publicly available for research purposes.[4] We compare Slope and Sloper to the state-of-the-art for univariate causal inference. These include Cure (Sgouritsa et al., 2015), IGCI (Janzing et al., 2012) and RESIT (Peters et al., 2014). From the class of ANM-based methods we compare to ANM-pHSIC (Hoyer et al., 2009; Mooij et al., 2016), which a recent survey identified as the most reliable ANM inference method (Mooij et al., 2016). We use the implementations by the authors, sticking to the recommended settings.

To run Slope, we have to define the parameter t, which is used to normalize the data X_i within a local component, on which the data Y_i is fitted. Generally, the exact value of t is not important for the algorithm, since it only defines the domain of the data points X_i, which can be compensated by the parameters of the fitted function. In our experiments, we use $t = 5$ and set the precision p for the parameters to three.

6.5.1 Evaluation Measures

As simply giving the *accuracy* over a set of experiments does not suffice to judge about the quality of an inference algorithm, we briefly explain frequently used measures. In general, it is not only important to have a high accuracy, but also to assign high confidence values to decisions about which the corresponding approach is most certain and low confidence values to less certain decisions as in our case high noise settings.

Commonly used measures to give more insight to this behaviour than the overall accuracy, are the area under the receiver operating characteristic (ROC) curve and the area under the precision recall (PR) curve. However, both have the drawback that they assign a preference to either select $X \to Y$ as the true positive and $Y \to X$ as the true negative or vice versa. As a consequence, they are not symmetric. The assignment of X and Y for the tested pairs is highly arbitrary and hence, the imposed preference of those tests is arbitrary, too.

An alternative measure is the accuracy with respect to the *decision rate*, which we simply denote by *accuracy curve*. The decision rate is the percentage of pairs for which we force a decision—i.e., a decision rate of $p\%$ means that we consider those $p\%$ of all decisions with the highest confidence. In contrast to ROC and PR the decision rate is independent of the label of the result. To get the accuracy curve, we simply calculate the accuracy per decision rate.

[4] `http://eda.mmci.uni-saarland.de/slope/`

Similar to ROC and PR, we can also calculate the area under the accuracy curve (AUAC), which is our preferred measure.

6.5.2 Synthetic Data

We first consider data with known ground truth. To generate such data, we follow the standard scheme of Hoyer et al. (2009). That is, we first generate X randomly according to a given distribution, and then generate Y as $Y = f(X) + N$, where f is a function that can be linear, cubic or reciprocal, and N is the noise term, which can either be additive or non-additive.

Accuracy

First, we evaluate the performance of Slope under different distributions. Following the scheme above, we generate X randomly from either

1. a uniform distribution with min $= -t$ and max $= t$, where $t \sim \text{Unif}(1, 10)$,
2. a sub-Gaussian distribution by sampling data with $\mathcal{N}(0, s)$, where $s \sim \text{Unif}(1, 10)$ and taking each value to the power of 0.7, keeping its sign,[5]
3. a binomial distribution with $p \sim \text{Unif}(0.1, 0.9)$ and the number of trials $t \sim \lceil \text{Unif}(1, 10) \rceil$, or
4. a Poisson distribution with $\lambda \sim \text{Unif}(1, 10)$.

Note that the binomial and Poisson distribution generate discrete data points, which with high probability results in duplicate values. To generate Y we first apply either a linear, cubic or reciprocal function on X, with fixed parameters, and add either additive noise using a uniform or Gaussian distribution with $t, s \sim \text{Unif}(1, \max(x)/2)$ or non-additive noise with $\mathcal{N}(0,1) |\sin(2\pi\nu X)| + \mathcal{N}(0,1) |\sin(2\pi(10\nu)X)|/4$ according to (Sgouritsa et al., 2015), where we choose $\nu \sim \text{Unif}(0.25, 1.1)$. For every combination we generate 100 data sets of 1 000 samples each.

Next, we apply Slope, RESIT, and IGCI and record their accuracies. Since all tested functions can be modelled by Slope, they can also be modelled by Sloper. Hence, the performance of Sloper is identical, and we only give the results for one of them. As they take up to hours to process a single pair, we do not consider Cure and ANM here. We give the averaged results over all three function types in Figure 6.2. In general, we find that Slope and IGCI perform on par and reach 100% for most setups, whereas Slope performs better on the sub-Gaussian data. If we consider the single results for linear, cubic and reciprocal, we find that on the linear data with sub-Gaussian distributed X, Slope performs on average 7% better than IGCI.

[5] We consider sub-Gaussian distributions since linear functions with both X and N Gaussian distributed are not identifiable by ANMs (Hoyer et al., 2009).

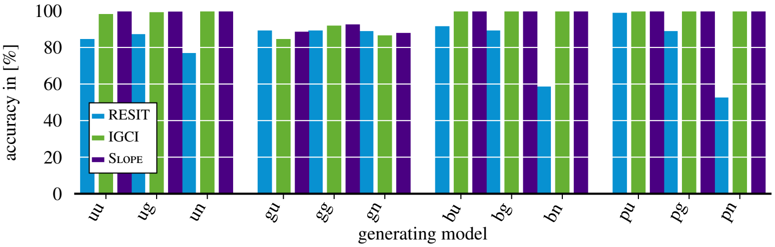


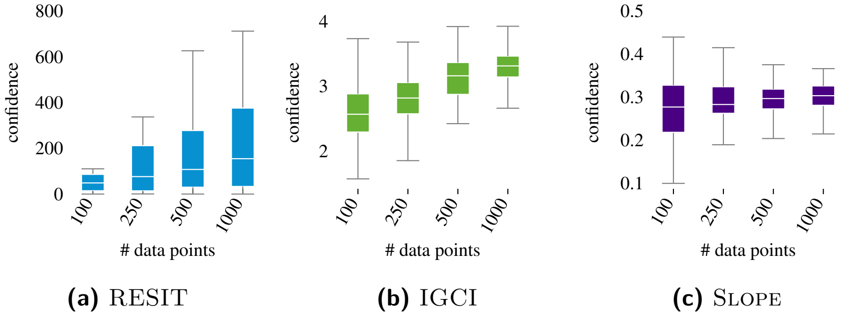
Figure 6.2: [Higher is better] Accuracies of Slope, RESIT and IGCI on synthetic data–Sloper performs identical to Slope. The first letter of the labels corresponds to the distribution of X (u: uniform, g: sub-Gaussian, b: binomial and p: Poisson), the second letter to that of the noise (u: uniform, g: Gaussian and n: non-additive).

Confidence

Second, we investigate the dependency of the RESIT, IGCI, and Slope scores on the size of the data. In an ideal world, a confidence score is not affected by the size of the data, as this allows easy comparison and ranking of scores.

To analyze this aspect, we generate 100 datasets of $100, 250, 500$ and $1\,000$ samples each, where X is Gaussian distributed and Y is a cubic function of X with uniform noise. Subsequently, we apply RESIT, IGCI and Slope and record their confidence values. We show the results per sample size in Figure 6.3. As each method uses a different score, the scale of the Y-axis is not important. What is important to note, is the trend of the scores over different sample sizes. We see that the mean of the confidence values of Slope is very consistent and nearly independent of the number of samples. In addition, our score becomes more precise with more data: the size of the confidence interval decreases. In strong contrast, the standard deviation of the confidence values increases with larger sample size for RESIT. For IGCI, we observe that the average confidence increases with the number of samples.

In addition to theses plots, we check if there is a significant mean shift in the confidence values for different sample sizes. Hence, we apply the exact two-sided Wilcoxon rank-sum test (Wilcoxon, 1945; Marx et al., 2016). In particular, we compare the confidence values for the sample sizes $100, 200, 500$ to the ones for sample size $1\,000$ for all methods. As result, we observe that for a significance level of 0.01 we find a significant shift in all three tests for IGCI. Also, for RESIT, there is a significant mean shift between the values for 100 and $1\,000$ as well as for 250 and $1\,000$. Slope is consistent from 250 samples onwards. In other words, while it is easy to compare and rank Slope scores, this is not the case for the two others—which, as we will see below results in comparatively bad accuracy curves.



(a) RESIT **(b)** IGCI **(c)** SLOPE

Figure 6.3: [The more stable the better] Confidence values on a cubic function for different sample sizes. Unlike RESIT and IGCI, the SLOPE scores can be meaningfully compared between different sample sizes.

Identifiability of ANMs on Synthetic Data

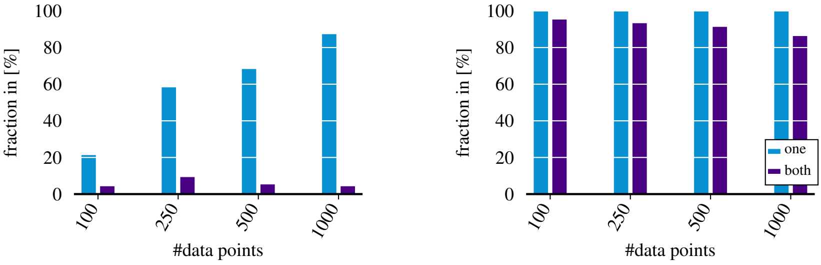
Connected to the vulnerability of the p-values of RESIT to the size of the data, we investigate in a similar problem. When the data size or the complexity of the function increases, the test for independence between X and N is likely to hold in both directions. Accordingly, we generate uniform data with additive Gaussian noise for different data sizes and plot the results for linear and cubic functions in Figure 6.4. We can observe that this problem does very rarely occur for the linear data. For the more complex generative function, the cubic function, we observe that quite frequently both directions are flagged as ANMs. Notably, most of the time one direction is significant, the other is so, too. In such cases, RESIT and other ANM based algorithms, decide for the more extreme p-value. As stated by Anderson et al. (2000), a more extreme p-value does not necessarily imply a stronger *independence*. The only valid statement we can make is that it is highly unlikely that the noise is *dependent* on X as well as on Y for the inverse direction. Deciding for the correct direction, however, is not well defined. Especially, if we consider that the p-values can be very low and in the order of 10^{-100}, as we saw in the previous experiment.

Since SLOPE does not rely on p-values, but decides based on the fit as well as the complexity of the model, we can avoid these problems.

Fitting Local Models

Local regression on duplicates in the data adds to the modelling power of SLOPE, yet, it may also lead to overfitting. Here we evaluate whether MDL protects us from picking up spurious structure.

To control non-determinacy, we sample X uniformly from k equidistant values over $[0, 1]$, i.e., $X \in [\frac{0}{k}, \frac{1}{k}, \cdots, \frac{k}{k}]$. To obtain Y, we apply a linear



(a) linear

(b) cubic

Figure 6.4: Percentage of cases where one (blue) or both (purple) causal directions are significant under an ANM.

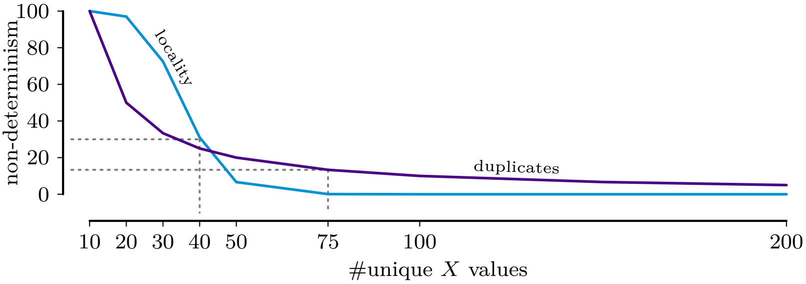


Figure 6.5: [SLOPE does not overfit] Percentage of local models SLOPE chooses, resp. the expected number of Y values per X value.

function and additive Gaussian noise as above. Per data set we sample 1 000 data points. In Figure 6.5 we plot the locality of the model, i.e., the average number of used local functions divided by the average number of bins SLOPE could have used, against the number of distinct X values. As a reference, we also include the average number of values of Y per value of X. We see that for at least 75 unique values, SLOPE does not use local models. Only at 40 distinct values, i.e., an average of 25 duplicates per X, SLOPE consistently starts to fit local models. This shows that if anything, rather than being prone to overfit, SLOPE is conservative in using local models.

GP Simulated Data

Next, we want to show that considering a richer function class is beneficial for our approach. As a showcase, we apply both SLOPE and SLOPER to the synthetic data pairs proposed by Mooij et al. (2016), where both the data over

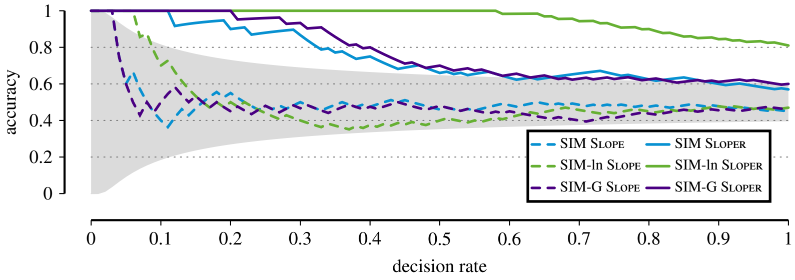


Figure 6.6: [Higher is better] Accuracy curves of Slope and Sloper on the SIM, SIM-ln and SIM-G data sets. The gray area refers to the 95% confidence interval of a coin flip.

the cause X and the function that maps X to Y have been generated using a Gaussian process (GP). We consider three scenarios,[6] each containing 100 pairs of 1 000 samples. The first one, *SIM* is the standard setup, *SIM-ln* has low noise levels and for *SIM-G* both the distribution of X as well as the additive noise variable are near Gaussian.

In Figure 6.6 we provide the accuracy curves for Slope and Sloper. Overall, we can observe that Sloper clearly improves upon the the results of Slope, since it is able to fit the more complex GP functions better. Especially for the low noise scenario, Sloper improves significantly and reaches an overall accuracy of 80%. In general, we can observe that the accuracy curves for both are good since the correct decisions have the highest confidence values.

If we consider the area under the accuracy curve, Sloper performs well having an AUAC of 96% on SIM-ln, 77% on SIM-G and 75% on SIM whereas Slope has an AUAC of about 50% for all of them. As we expect our approach to work better in a low noise setup since a) in this case it will be easier to fit non-linear regression functions and b) linear-Gaussian functions are generally not identifiable (Peters et al., 2011b), it is not surprising that Sloper performs best on the SIM-ln data set.

6.5.3 Real World Data

Next, we evaluate Slope on real world benchmark data. In particular, we consider the Tübingen cause-effect data set.[7] At the time of writing the data set included 98 univariate numeric cause effect pairs. We first compare Slope to IGCI, RESIT, ANM, and Cure, using the suggested parameter settings for this benchmark. Afterwards, we compare different variants of Slope.

[6] We exclude the confounded scenario since it violates our assumptions.

[7] https://webdav.tuebingen.mpg.de/cause-effect/

Accuracy Curves and Overall Accuracy

We first consider the overall accuracy and the accuracy curves over the benchmark data, where we weight all decisions according to the weights specified in the benchmark. In case an algorithm does not decide, we consider this a toss-up and weight these results as one half of the corresponding weight.

We plot the results in Figure 6.7, where in addition we show the 95% confidence interval for the binomial distribution with success probability of 0.5 in gray. We observe that Slope strongly outperforms its competitors in both area under the accuracy curve and overall accuracy; it identifies the correct result for top-ranked 34 data sets, over the top-72 pairs (which correspond to 72.4% of the weights) it has an accuracy of 90%, while over all pairs it obtains an accuracy of 81.7%.

In Figure 6.8 we show the corresponding confidence values of Slope for the benchmark pairs. The plot emphasizes not only the predictive power of Slope, but also the strong correlation between confidence value and accuracy. In comparison to the other approaches the area under the accuracy curve (Figure 6.7) of Slope is stable and only decreases slightly at the very end. Our competitors, obtain overall accuracies of between 56% (Cure) and 71% (RESIT), which for the most part are insignificant with regard to a fair coin flip. This is also reflected in the AUAC values, which lie between 0.588 (Cure) and 0.736 (IGCI), whereas Slope has an AUAC of 0.942.

If we not only consider the confidence values, but also our proposed statistical test based on the absolute difference, we can improve our results even further. After adjusting the p-values using the Benjamini-Hochberg correction (Benjamini and Hochberg, 1995) to control the false discovery rate (FDR), 81 out of the 98 decisions are significant w.r.t. $\alpha = 0.001$. As shown in Figure 6.8 the pairs rated as insignificant correspond to small confidence values. In addition, from the 17 insignificant pairs, 11 were inferred incorrect from Slope and 6 correct. Over the significant pairs the weighted accuracy increases to 85.2%, and the AUAC to 0.965.

To provide further evidence that the confidence values and the p-values are indeed related, we plot the adjusted p-values and confidence values in Figure 6.9(a). We observe that high confidence values correspond to highly significant p-values. We also computed the area under the accuracy curve for Slope when ranking by p-values, and find it is only slightly worse than that ranked by confidence. We posit that confidence works better as it is more independent of the data size. To test this, we calculate the correlation between data size and corresponding measures using the maximal information coefficient (MIC) (Reshef et al., 2011). We find a medium correlation between confidence and p-values (0.64), and between p-values and data size (0.55), and only a weak correlation between confidence and data size (0.31).

Apart from the accuracies, we also tracked which functional dependencies

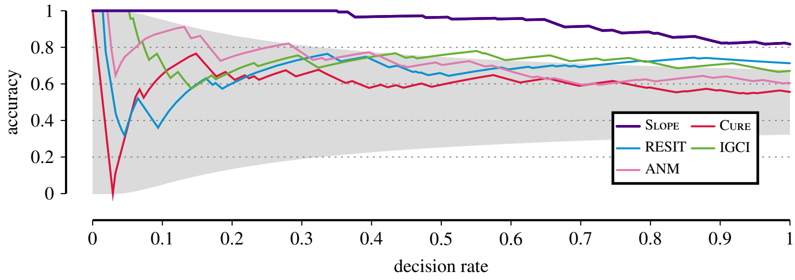


Figure 6.7: [Higher is better] Accuracy curves of SLOPE, CURE, RESIT, IGCI and ANM on the Tübingen benchmark data set (98 pairs).

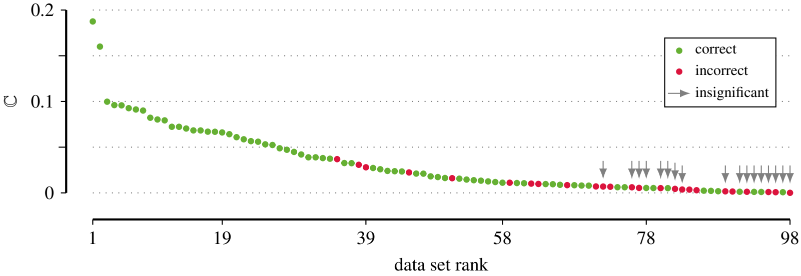


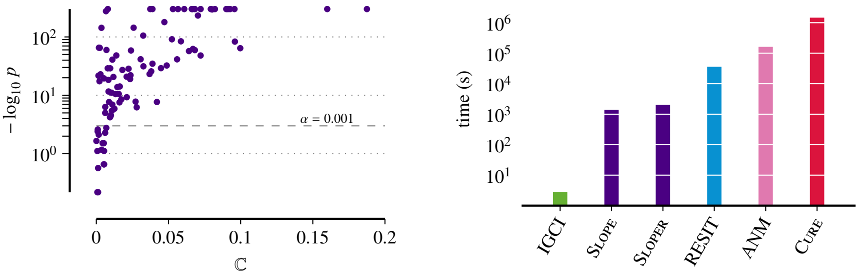
Figure 6.8: Confidence values for SLOPE for the Tübingen benchmark pairs, in descending order, corresponding to Figure 6.7. We marked correct inferences in green, errors in red, and inferences insignificant at $\alpha = 0.001$ for the absolute p-value test are marked with a gray arrow on top.

SLOPE found on the benchmark data. We found that most of the time (54.6%), it fits linear functions. For 23.7% of the data it fits exponential models, and for 15.5% cubic models. Quadratic and reciprocal models are rarely fitted (6.2%).

A key observation to make here is that although we allow to fit complex models, in many cases SLOPE prefers a simple model as it has sufficient explanatory power at lower model costs. In fact, if we only allow linear functions SLOPE is only a few percentage points less accurate compared to the full class of functions. The confidence of the method, however, is much larger in the latter case as only then SLOPE is able to better measure the difference in complexity in both directions.

Relative p-values

Next, we compare the absolute p-value that we applied in the last section, to finding a cut-off for the confidence value based on the relative significance test.



(a) $\mathcal{C}$ vs. p-value

(b) runtime

Figure 6.9: Left: Confidence and significance of Slope on the Tübingen benchmark pairs. Only samples with low confidence are also insignificant. Right: Runtime in seconds over all 98 pairs, in log-scale. Slope and Sloper both are more accurate than all, and faster than all except for IGCI.

As explained in Section 6.2, the confidence value can be interpreted as a relative p-value with respect to a given reference size, e.g. 1 000 bits. Although ranking by relative p-value would obviously result in the same area under the accuracy curve as ranking by confidence value, it does allow us to determine a sensible threshold to decide between significant and random decisions.

Given budget $b = 1\,000$ bits and a significance level $\alpha = 0.05$, we obtain a confidence threshold $t_{\mathcal{C}} = 0.00864$. If we reconsider Figure 6.8, we observe that 32 decisions are rendered insignificant by this threshold. From those, 17 are incorrect and 15 correct. Consequently, this threshold exactly prevents our algorithm to make $50 : 50$, or random decisions. At the same time, considering only the significant decisions, results in an accuracy of 95.2%. Alternatively, if we lower the significance threshold to 0.01, eleven more decisions are insignificant, out of which more than two thirds are correct, which implies that a $\alpha = 0.01$ might be too restrictive.

Area under the ROC, PR and Accuracy Curve

In this paragraph, we briefly discuss the different evaluation measures as the area under the ROC, PR, and accuracy curve. We use each measure to evaluate Slope, Sloper, Cure, RESIT, IGCI and ANM on both the Tübingen data set including 98 univariate pairs and an older version (version 0.9, on which many of our competitors were evaluated), including only 79 univariate pairs. For the ROC and PR curves, we compute both directions, where ROC_X corresponds to selecting $X \rightarrow Y$ as true positive and ROC_Y to selecting $Y \rightarrow X$ as true positive—accordingly so for PR. We show the results in Table 6.1.

Table 6.1: [Higher is better] Area under the ROC, PR and Accuracy curves for Slope, Sloper, Cure, RESIT, IGCI and ANM on both the Tübingen data set including 98 univariate pairs and the older version 0.9, including only 79 univariate pairs. All decisions are weighted with the corresponding weights of the benchmark.

		Slope	Sloper	Cure	RESIT	IGCI	ANM
Tübingen$_{98}$	ROC$_X$	**0.898**	0.865	0.424	0.573	0.671	0.472
	ROC$_Y$	**0.897**	0.862	0.413	0.564	0.675	0.472
	PR$_X$	**0.962**	0.948	0.716	0.791	0.808	0.734
	PR$_Y$	**0.728**	0.705	0.232	0.265	0.600	0.255
	AUAC	**0.942**	0.927	0.588	0.676	0.736	0.713
Tübingen$_{79}$	ROC$_X$	**0.812**	0.792	0.381	0.508	0.388	0.469
	ROC$_Y$	**0.851**	0.830	0.414	0.528	0.422	0.502
	PR$_X$	**0.942**	0.935	0.740	0.800	0.675	0.742
	PR$_Y$	**0.575**	0.573	0.200	0.254	0.269	0.232
	AUAC	**0.933**	0.924	0.819	0.534	0.715	0.802

First of all, we observe that both Slope and Sloper have very similar results and outperform the competing approaches on each scoring metric. Further, we observe that it makes a huge difference for every approach whether we consider PR$_X$ or PR$_Y$ (see e.g. Cure). This difference relates to the imbalance of the benchmark data set, in which the majority of the data pairs have X as the true cause. However, since it is an arbitrary choice how to assign X and Y for each data set, we consider the area under the precision recall curve not as an appropriate measure for causal inference. We observe a similar effect for the area under the ROC curve, but much weaker. Still the score is dependent on the choice of the true positive and hence we consider the area under the accuracy curve as the most objective measure.

Slope and its Variations

As a last test on the benchmark data set, we compare Slope, Sloper and two additional variants of our algorithm. Those are Slope$_G$, which only fits global functions and Slope$_P$, which has the same results as Slope, but uses the absolute p-value as confidence. For each variant we plot the accuracy curve for all significant decisions with respect to $\alpha = 0.001$ in Figure 6.10.

First of all, we observe that Sloper is on par with Slope up to a decision rate of 75% and reaches an overall accuracy of 80%. The AUAC of Sloper (0.936) is nearly as good as the one for Slope (0.945). Hence, only in the low confidence region, Sloper had a slightly worse performance. When we

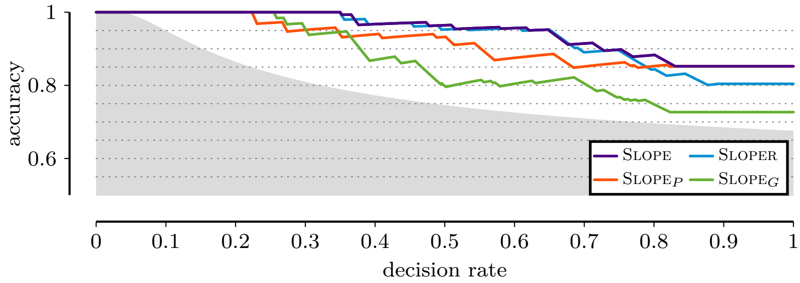


Figure 6.10: [Higher is better] Accuracy curves for SLOPE, SLOPER, SLOPE$_P$ and SLOPE$_G$ on the Tübingen benchmark data set. SLOPE$_P$ is inferred with SLOPE, but ranked according to the p-value and SLOPE$_G$ only fits the data with a single global function. Only significant decisions with respect to $\alpha = 0.001$ are considered.

inspected those decisions, we found that the corresponding pairs mainly consisted of pairs with high noise levels. This explains why SLOPE and SLOPER made different decisions as both were not very certain. Moreover, we observe that using the p-value as confidence measure leads to a slightly worse accuracy curve and AUAC of 0.918, however, as expected it is still good as the confidence values correlate with the p-values. SLOPE$_G$ has an overall accuracy of about 73% and an AUAC of 0.861, which clearly shows that fitting local functions boosts the accuracy significantly.

RUNTIME

Next, we evaluate the computational efficiency of SLOPE. To this end we report, per method, the wall-clock time needed to decide on all 98 pairs of the benchmark data set. We ran these experiments on Linux servers with two six-core Intel Xenon E5-2643v2 processors and 64GB RAM. The implementations of SLOPE, IGCI and RESIT are single-threaded, whereas ANM and CURE are implemented in Matlab and use the default number of threads. We give the results in Figure 6.9. We see that IGCI is fastest, followed by SLOPE and SLOPER, taking 1 475 respectively 1 936 seconds to processes all pairs. The other competitors are all at least one order of magnitude slower. Taking 13 days, CURE has the longest runtime. The large gain in runtime of SLOPE compared to RESIT, ANM and CURE rises from the fact that those methods employ Gaussian process regression to fit the functions.

6.5.4 Case Study: Octet Binary Semi Conductors

To evaluate real-world performance we conduct a case study on octet binary semi-conductors (Ghiringhelli et al., 2015; Van Vechten, 1969). In essence, the data set includes the 82 different materials one can form by taking one each from two specific groups of atoms, and of which the resulting material either forms a rocksalt or zincblende crystal structure. The aim of current research is to predict, given a set of physical properties, the crystal structure of the material. A key component to distinguish between both forms is the energy difference δ_E between rocksalt and zincblende. At the time of writing, it is not known which combination of physical properties can be used to calculate δ_E, however, there exist candidates that are known to have some impact (Ghiringhelli et al., 2015; Goldsmith et al., 2017). Since the data set contains very high quality measurements, it is well suited as a case study for our method.

In particular, form the set of physical properties, which also contains derived properties consisting of combinations or log transformations, we extracted the top 10 that had the highest association to δ_e (Mandros et al., 2017). The point is that we know that all of these properties somehow influence δ_E, but an exact formula to calculate δ_E is not known yet. After consulting the domain experts, we thus obtain 10 new cause effect pairs. For each of those pairs, we define δ_E as X and one of the top 10 features as Y. Since the energy difference is influenced by the features, we can assume that $Y \to X$ is the true causal direction for all pairs. For more detailed information to the data set, we refer to Ghiringhelli et al. (2015). We make these extracted cause-effect pairs available for research purposes.[8]

Last, we applied Slope, Sloper and their competitors to each of the ten pairs. As result, we find that Slope and Sloper perform identical and infer the correct direction for 9 out of 10 pairs. The only error is also the only insignificant score ($p = 0.199$) at $\alpha = 0.001$. In comparison, we find that Cure infers all pairs correctly, whereas IGCI makes the same decisions as Slope. RESIT and ANM, on the other hand, only get 4 resp. 5 pairs correct.

6.6 Conclusion

We studied the problem of inferring the causal direction between two univariate numeric random variables X and Y. To model the causal dependencies, we proposed an MDL-based framework employing local and global regression. Further, we proposed Slope, an efficient linear-time algorithm, to instantiate this framework. To fit more flexible functions, we extended Slope to consider combinations of basis functions. Moreover, we introduced ten new cause-effect pairs from a material science data set.

[8] http://eda.mmci.uni-saarland.de/slope/

Empirical evaluations on synthetic and real world-data show that Slope reliably infers the correct causal direction with high accuracy. On benchmark data, at 82% accuracy, Slope outperforms the state-of-the-art by more than 10% and has a more robust accuracy curve while additionally also being computationally more efficient.

Despite the excellent performance on synthetic and benchmark data, we did not focus much on another critical aspect of causal inference: identifiability. Ideally, we would like to provide conditions under which we are certain that we can infer the correct causal direction; in the sample limit. In the following chapter, we will provide exactly this result and show that we can guarantee identifiability if we use lossy (or sloppy) MDL encodings.

Chapter 7

Identifiability of Cause and Effect using Regularized Regression

In the previous chapter, we considered the problem of inferring the causal direction between two correlated numeric random variables under the assumption of acyclicity and causal sufficiency. We proposed a two-part MDL score, which models dependencies using local and global regression functions to approximate the algorithmic independence of conditionals. Although the corresponding algorithm, SLOPE, shows a good performance on benchmark data sets known at that time, one point of criticism is that SLOPE has no guarantees with regard to identifiability. In this chapter, we aim at solving this problem.

As identifiable, we consider models that, under specific conditions and given infinite data, are *guaranteed* to infer the correct causal direction. After all, unless we know we can unambiguously distinguish cause from effect given infinitely many samples, there is little point in trying it on fewer samples. A lot of research in causal inference is therefore focused on identifiability results, figuring out the conditions under which models are identifiable (Peters et al., 2011b). Most well researched in this regard are additive noise models, which are known to be identifiable for many different setups, such as linear functions with non-Gaussian noise (Shimizu et al., 2006) or non-linear functions with additive noise (Hoyer et al., 2009). For a broader overview, we refer to Section 6.4. Nonetheless, ANMs are a rather strict assumption on how the world works;

This chapter is based on Marx and Vreeken (2019c).

in practice, we often find functions and independent noise for both directions, which puts us back at square one.

In this chapter, we aim at combining two ideas to get an identifiable model, which performs well in practice. The first part is to follow the general approach to use regularized regression as done in Slope or also in CAM (Bühlmann et al., 2014), where the idea is to infer a hierarchy among a set of random variables based on a log-likelihood score. As the second part, we build upon a recent result, which shows that given the true generating function, the causal direction is identifiable by simply comparing the residual error (Blöbaum et al., 2018). That is, if we fit both $Y = f(X) + N_X$ and $X = g(X) + N_Y$ such that f and g minimize the respective residual errors N_X and N_Y, the expected squared error in the causal direction is smaller than for the anti-causal direction. The limitation of this approach is, however, that the two functions we fit should be of a similar complexity class to avoid comparing residuals of arbitrarily under- or overfit models. Thus, we need to select a specific, quite restricted model class beforehand, e.g. polynomials up to degree three, and hope that this class of functions can model the true function.

To benefit from both ideas, we propose a very light-weight assumption, that is, we assume that the best anti-causal model requires at least as many parameters as the causal model. As a consequence, we can derive identifiability guarantees for a large class of L_0-regularized regression functions. We carefully justify this assumption by showing that it is connected to the algorithmic model of causality and, in addition, holds trivially for a vast class of generating functions. As a proof of concept, we instantiate this general framework using spline-based regression and show that our approach performs well even if not all of our assumptions are met.

The roadmap of this chapter is as follows. First, in Section 7.1, we give a brief introduction to the main concepts of RECI (Blöbaum et al., 2018) that we build upon. We then, in Section 7.2, derive the key assumption to our approach and connect it to the AIC postulate. Based on this assumption, we show that the class of *identifiable regression-based scoring functions* is identifiable in Section 7.3 and show how to instantiate it. In Section 7.4, we provide the Sloppy algorithm to compute our causal indicator in practice and empirically evaluate Sloppy in Section 7.5.

7.1 Causal Inference by Regression Error

In this section, we briefly explain the main idea behind RECI (Blöbaum et al., 2018), its limitations, and how we want to solve them.

Similar to the previous chapter, we consider causal inference based on an i.i.d. sample of two univariate continuous random variables X and Y and assume acyclicity and causal sufficiency. Further, we write β_f for the set of parameters of a function f and denote with $\|\beta_f\|_0$ to the number of non-zero parameters.

7.1.1 A brief Introduction to RECI

The general idea behind RECI (Blöbaum et al., 2018) is that we can infer cause from effect simply by comparing the regression error of the best fitting model for the causal and anti-causal direction. In particular, they formulate a set of assumptions under which they can differentiate between cause and effect with certainty. Formally, if ϕ is the function that minimizes the least-squared error when predicting the effect Y from the cause X and vice versa ψ the function minimizing the error when predicting the cause from the effect, Blöbaum et al. (2018) formulate a set of assumptions, under which

$$\mathbb{E}[(Y_\alpha - \phi(X))^2] \leq \mathbb{E}[(X - \psi(Y_\alpha))^2] \tag{7.1}$$

always holds. In other words, when their assumptions hold, Blöbaum et al. (2018) prove that we can *identify* the true causal model using Equation (7.1). Hence, we can use the asymmetry in the regression error to infer the causal direction between two random variables.

Identifiability is an important concept in causal inference, as we can only make statements about the true causal model when we can guarantee identifiability. As this cannot be done in general, the goal is to prove identifiability under a set of assumptions that are as lightweight and as general as possible. We state the main assumptions for RECI below (Blöbaum et al., 2018).

Assumption 7.1 (Causal model) *We can write the effect as*

$$Y_\alpha := f(X) + \alpha N\,,$$

with noise term N and parameter α restricting the noise level.

Assumption 7.2 (Unbiased noise) *The noise term N is unbiased and has unit variance.*

Assumption 7.3 (Compact supports) *The distribution of X has compact support and w.l.o.g. X attains values between 0 and 1, which can be achieved by normalizing X. Further, the distribution of N has compact support and there exist values $n_+ > 0 > n_-$ such that for each value $x \in \mathcal{X}$, $[n_-, n_+]$ is the smallest interval that containing the support for the conditional density of N given x. Hence, we know that $[\alpha n_-, 1 + \alpha n_+]$ is the smallest interval containing the support of the density of Y_α and rescale it to*

$$\tilde{Y}_\alpha := \frac{Y_\alpha - \alpha n_-}{1 + \alpha n_+ - \alpha n_-}\,.$$

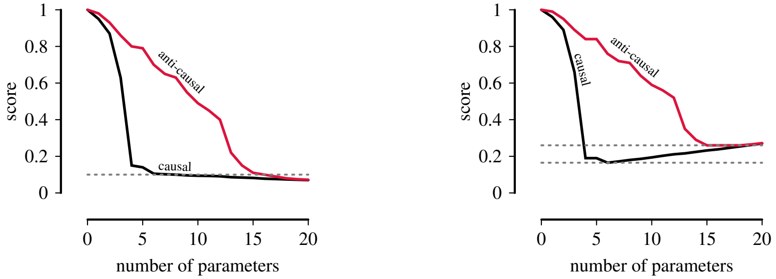


Figure 7.1: Left: Error for the best fitting function in the causal and anti-causal direction, when restricting the number of parameters. Right: Error plus L_0 penalty on the number of parameters for the same data.

$\tilde{Y}_\alpha$ has the same scale as X and attains values between 0 *and* 1*.*

Based on Assumptions 7.1-7.3, Blöbaum et al. (2018) show that their approach works under the additive noise assumption, that is, the cause X is independent of the noise term. In addition, their framework allows slight violations of this assumption, as it is also valid when there is a low dependence between the noise and the cause. In particular, they show that Equation (7.1) holds for strictly monotonically increasing and twice differentiable ϕ and trivially holds for non-invertible functions, as for the latter case, there is an information loss in the anti-causal direction. Last, they show that Equation (7.1) holds with equality if and only if ϕ is a linear function, which means that we cannot identify the causal direction for linear functions.

In general, RECI provides a solid framework to identify cause from effect only based on regression error, which is easy to obtain. Also, Assumption 7.3 is not very restrictive, as we can achieve it by normalizing the data, if we have a sufficient number of samples. The problem is, however, that in practice we do not know the true functions. Hence, we need to restrict ourselves to comparing functions of the same type, i.e., polynomials of degree three or six, but cannot compare across functions of different complexities. The goal of this work is to solve precisely this issue while conserving the identifiability guarantees.

7.1.2 Main Idea

The key idea to overcome the limitations of RECI is simple. Instead of only comparing the regression error, we use regularized regression and compare the regularized scores of those functions for which they are minimal. To illustrate this, consider the following example.

Example 7.1 *Assume we are given an i.i.d. sample from the joint distribution of X and Y, and we know the true causal direction. Now we use our go-to algorithm to fit a regression function in both the causal and the anti-causal direction. In Figure 7.1, we plot the minimum regression error for both directions, where we gradually allow the model to fit more parameters. If we allow a sufficient number of parameters, we can reduce the regression error for both directions to approximately the same level. As a consequence, only comparing the regression errors of the best fitting model does not identify the correct causal direction. However, we observe that we can find a much simpler function for the true causal direction, which attains approximately the same regression error; in contrast to the anti-causal direction. When we compare the scores of those functions minimizing the regression error plus an L_0 penalty over the parameters (right plot), we can identify the correct model, as there is a clear difference between both scores.*

Of course, we do not want to rely on a proof by an artificial example, but from Example 7.1 we get our motivation. What we completely forgot about for a second, is the identifiability. It is known that we can use regularized regression to fit functions, but does this also result in scoring functions that are identifiable?

In the following, we will show that it does. In particular, we define a class of scoring functions for regularized regression that are identifiable under the assumption that the mechanism mapping the cause to the effect is at most as complex as the anti-causal one. We derive and justify this assumption from the algorithmic model of causality using Kolmogorov's structure function.

7.2 Principled Regularization for Causality

To define our new inference rule, we need to introduce one more assumption, that is, we assume that true causal model is not more complex than the anti-causal model. This claim might not be too intuitive and hence we are going to carefully justify our assumption in the following from the algorithmic model of causality, which we will briefly recap.

In the following, we will, as common when talking about Kolmogorov complexity (see Definition 3.1), use lower case letters such as x to refer to the binary string representation of X. As stated in Postulate 5.2, the algorithmic model of causality for two random variables X and Y with ground truth graph $X \rightarrow Y$ can be summarized as follows. If x is the cause, we generate y as

$$y = q(x, n) \ ,$$

where n is a noise variable and q is a program of constant complexity that is independent of its inputs. This program can in theory model every physical process (Deutsch, 1985), which includes functional relationships. Hence, it also supports the causal model that we assume in this chapter (Assumption 7.1).

Under the algorithmic model of causality, we can further assume that the algorithmic independence of conditionals postulate holds (Janzing and Schölkopf, 2010) and thus can infer the causal direction by comparing the algorithmic descriptions of the factorizations of P_{XY} (see Postulate 5.3). In particular, if $X \to Y$, we have that

$$K(P_X) + K(P_{Y|X}) \overset{+}{\leq} K(P_Y) + K(P_{X|Y})\,. \tag{7.2}$$

That is, we infer that direction, which provides the simplest factorization of the joint distribution of X and Y. In theory, we could infer the causal direction for any physical process—if only we could compute Kolmogorov complexity. One way to approximate Kolmogorov complexity is to split it into the complexity of the meaningful information that can be efficiently represented by a short program and the complexity of the irreducible noise that cannot be modelled efficiently. A sound theoretical concept that differentiates between those quantities is described by Kolmogorov's structure function.

7.2.1 Kolmogorov's Structure Function

Although there is no written publication of Kolmogorov about the structure function, it has found its way into research (Vereshchagin and Vitányi, 2004). The key concept we need is that of a model $S \ni x$, that is, a model is a set of binary strings of which x is a member. Given such a set S and no further input, we will need $\log |S|$ bits to look up x in S.[1] Simple models, i.e., those with low Kolmogorov complexity $K(S)$, will consist of many possible strings and hence it will take us relatively many bits to identify x in S. If we increase the budget for $K(S)$ we can contemplate more complex models that consist of fewer possible strings, and for these it will cost much fewer bits to single out x. In the most extreme case, where we set $K(S) = K(x)$, we can have $S = \{x\}$. Formally, we can describe this relationship as Kolmogorov's structure function

$$h_x(\alpha) = \min_S \{\log |S| \colon S \ni x, K(S) \leq \alpha\}$$

with S being a contemplated model for x and α a non-negative number bounding the complexity of the contemplated S's (Vereshchagin and Vitányi, 2004). There exists complexity thresholds α for which $\alpha + h_x(\alpha) = K(x) + \mathcal{O}(1)$. For these, the associated model S is called an *optimal set* for x. Its description of up to α bits is called *sufficient statistic* for x. Moreover, for a sufficient statistic S it holds that $K(x) \leq K(S) + \log |S| \leq K(x) + \mathcal{O}(1)$. If we consider all

[1] As usual, we use log with base 2 to refer to bits.

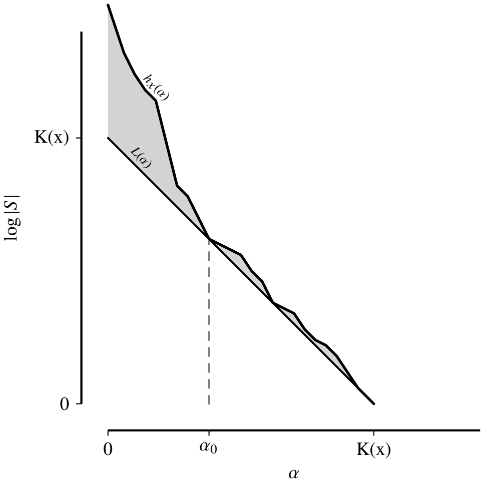


Figure 7.2: Shown is the sufficiency line $L(\alpha) = K(x) - \alpha$ and the structure function h_x. At α_0 corresponding to the minimum sufficient statistic, h_x hits the sufficiency line for the first time. For all $\alpha > \alpha_0$ where $h_x(\alpha) = L(\alpha)$, the corresponding set S is a sufficient statistic for x. Marked in red it the gap between $K(x)$ and $h_x(\alpha) + \alpha$.

sufficient statistics S for x, we call that S which is associated with the smallest α—i.e., α_0, the *minimal sufficient statistic* for x. That is, the minimum sufficient statistic S contains all meaningful information about x and the associated term $h_x(\alpha_0)$ measures the complexity of the irreducible noise contained in x. Further, it holds that $h_x(\alpha_0) + \alpha_0 = K(x)$.

In Figure 7.2 we visualize this concept as suggested by Vereshchagin and Vitányi (2004). We see that for α_0 the structure function h meets the *sufficiency line*, that is defined as $L(\alpha) = K(x) - \alpha$, which is optimal and hence $\alpha_0 + h_x(\alpha_0) = K(x)$. For $\alpha < \alpha_0$, $h_x(\alpha)$ can be arbitrarily far above the sufficiency line and for $\alpha > \alpha_0$, $h_x(\alpha)$ is within a constant term above the sufficiency line.

Similar to conditional Kolmogorov complexity, we define the conditional structure function as

$$h_x(i \mid y) = \min_S \{\log |S| \colon S \ni x, K(S \mid y) \leq i\} \, .$$

We will need this conditional version as we will be considering functional relationships from X to Y and vice versa.

Now let us consider Equation (7.2) again and let x and y correspond to the binary string representations of X and Y. Further, be i_0^x the complexity level of the minimum sufficient statistic of x conditioned on y, and accordingly i_0^y the complexity level of the minimum sufficient statistic for y given x. We can rewrite Equation (7.2) as

$$K(P_X) + i_0^y + h_y(i_0^y \mid x) \leq K(P_Y) + i_0^x + h_x(i_0^x \mid y)\,.$$

In the following, we explain the above inequality given that Assumption 7.1-7.3 hold. As i_0^x contains all meaningful information of x given y, $h_x(i_0^x \mid y)$ relates to $K(C - \psi(E_\alpha))$ and $h_y(i_0^y \mid x)$ relates to $K(E_\alpha - \phi(C)) \approx K(N)$. Further, we find according to Postulate 5.2, that the program modelling the causal function (which relates to i_0^x) has constant complexity. For invertible functions, it is likely that the same holds for the function in the anti-causal direction. If the variance of the noise term goes to zero, that is, the function is near deterministic, Blöbaum et al. (2018) showed that the expected error for the causal model is smaller or equal to the error in the anti-causal direction—i.e., $h_y(i_0^y \mid x) \leq h_x(i_0^x \mid y)$. Thus, we would essentially ignore the Kolmogorov complexities of the marginals, or assume that $K(P_X) \overset{+}{=} K(P_Y)$. For almost deterministic data, this assumption seems sensible, since in this case, we can compute y from x with a program of constant complexity. Under this premise, we can infer that $X \rightarrow Y$, if

$$i_0^y + h_y(i_0^y \mid x) \leq i_0^x + h_x(i_0^x \mid y)\,. \tag{7.3}$$

Note that this inequality also holds if the function ϕ is not invertible and there does not exist an inverse function ψ. This follows from the fact that there is an information loss in the anti-causal direction and we cannot efficiently use the information about x to derive y. In addition, we can see from Equation (7.3) that if we only consider the regression error, it is important to know the true functions. If we do not, and overfit, e.g., in the anti-causal direction, we fit noise and obtain lower errors than for the true function, which can lead to wrong inferences. Formally, if we allow for a complexity level $i^x > i_0^x$, it is possible and for large i^x will eventually happen that $h_x(i^x \mid y) < h_x(i_0^x \mid y)$. Similarly, if we allow for $i^y < i_0^y$, which relates to underfitting in the causal direction, we will end up making false decisions.

Hence, we need to include the complexity of the model into our score, without breaking the identifiability results. According to the algorithmic model of causality, the program describing the causal mechanism, which relates to i_0^y has constant complexity. If the function ϕ is invertible, the same will likely hold for the i_0^x. Otherwise, if the function is non-invertible, it is likely that $i_0^y \overset{+}{\leq} i_0^x$. As we cannot compute Kolmogorov complexity, we need to formalize this idea differently. In essence, if the causal mechanism has a lower complexity than the anti-causal one, the true causal function ϕ should need at most as many parameters or degrees of freedom as the reverse function ψ. We formulate this in Assumption 7.4.

Assumption 7.4 (Simplicity) *Let Y_α be generated as in Assumption 7.1. Further, let ϕ be the function minimizing the expected least-squared error for predicting the effect Y from the cause X and ψ be the function minimizing the expected least-squared error in the anti-causal direction. We assume that ψ has at least as many parameters as ϕ, i.e., $\|\beta_\phi\|_0 \leq \|\beta_\psi\|_0$.*

While we cannot show that Assumption 7.4 holds in general, there are strong indications that it holds in many real-world settings. For example, if we know that ϕ consists of a linear combination of basis functions that are linearly independent of each other, we cannot find an inverse function that has fewer degrees of freedom. Moreover, Kilbertus et al. (2018) recently considered the problem of anti-causal learning and give indications on why it is harder than learning the causal direction. In particular, they give various examples, why it is simpler to learn the causal direction, from which we selected a few. As for low degree polynomials, it is easy to see that it is not possible to formulate an inverse with less parameters as the original function, the Abel-Ruffini theorem states that general polynomial equations of degree greater than 4 do not have an algebraic solution (Abel, 1826). Further, it is known that some elementary transcendental functions as $X + \sin(X)$ do not have an elementary inverse. In addition, in cryptography there exist the concept of a one-way function (Abel, 1826). Those are functions, that are easy to obtain in one direction but almost impossible to reverse.

Utilizing Assumption 7.4, we can finally connect all the dots and introduce our new framework.

7.3 Identifiable Regularized Regression

In the following, we show how we can design scoring functions, which 1) allow to identify the true causal direction under Assumptions 7.1-7.4, 2) help to identify the true functions ϕ and ψ and 3) are more robust w.r.t. overfitting. To this end, we define below an *Identifiable Regression-based Scoring Function*, or short IRSF and show that an IRSF fulfills the claims listed above.

Definition 7.1 (IRSF) *Given two random variables X and Y and a regression function ϕ that maps X to Y. Further, we are given a scoring function $S : \mathbb{R}_{\geq 0} \times \mathbb{N} \mapsto \mathbb{R}$ that takes as input the expected least-squared error $\mathbb{E}[(Y - \phi(X))^2]$ and the number of parameters of ϕ, $\|\beta_\phi\|_0$. We call such a scoring function*

$$S(Y \mid X, \phi) := \gamma(\mathbb{E}[(Y - \phi(X))^2]) + \lambda(\|\beta_\phi\|_0)$$

an Identifiable Regression-based Scoring Function (IRSF), if both $\gamma : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$ and $\lambda : \mathbb{N} \mapsto \mathbb{R}$ are strictly monotonically increasing.

It is easy to see that the number of parameters corresponds to the complexity of the function and hence $\lambda(\|\beta_\phi\|_0)$ relates to i_0^y. Further, under Assumptions 7.1-7.2, we can see that $\gamma(\mathbb{E}[(Y-\phi(X))^2])$ can be formulated to approximate $h_y(i_0^y \mid x)$. However, the question that remains is, can we identify this model and under which conditions. This we formalize in our main theorem.

Theorem 7.1 *Let Assumptions 7.1-7.4 hold, where ϕ denotes the function that minimizes the expected least-squared error when predicting the effect Y from the cause X and ψ be the function minimizing the expected least-squared error for predicting X from Y—i.e., $\phi(x) = \mathbb{E}[Y|x]$ and vice versa $\psi(y) = \mathbb{E}[X|y]$. Further, let S be an IRSF according to Definition 7.1. The following limit always holds*

$$\lim_{\alpha \to 0} \frac{S(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2], \|\beta_\phi\|_0)}{S(\mathbb{E}[(X - \psi(\tilde{Y}_\alpha))^2], \|\beta_\psi\|_0)} \geq 1\,,$$

with equality if and only if ϕ is linear.

Proof: *We know from Blöbaum et al. (2018) that under Assumptions 7.1-7.3 the following always holds*

$$(*) = \lim_{\alpha \to 0} \frac{\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2]}{\mathbb{E}[(X - \psi(\tilde{Y}_\alpha))^2]} \geq 1\,. \qquad (7.4)$$

As S is an IRSF, we can write it as $S(a,b) := \gamma(a) + \lambda(b)$, where γ is a strictly monotonically increasing function. Hence, the statement does not change by applying γ to the nominator and denominator in Equation (7.4). *Based on Assumption 7.4 we know that $\|\beta_\phi\|_0 \leq \|\beta_\psi\|_0$. Hence,*

$$\frac{\gamma(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2]) + \|\beta_\phi\|_0}{\gamma(\mathbb{E}[(X - \psi(\tilde{Y}_\alpha))^2]) + \|\beta_\psi\|_0} \geq (*)\,,$$

with equality if and only if ϕ and ψ are linear and thus $\|\beta_\phi\|_0 = \|\beta_\psi\|_0 = 1$. As λ is strictly monotonically increasing, applying it to $\|\beta_\phi\|_0$ and $\|\beta_\psi\|_0$ will not change this statement. □

7.3.1 Specifying γ and λ

Theorem 7.1 holds independently of how we exactly specify γ and λ. The problem, however, is that we do not know ϕ nor ψ beforehand. If we knew those functions, we could also apply the inference rule that is used in RECI (Blöbaum et al., 2018). The advantage of our score is that it not only identifies the true causal direction, when given ϕ and ψ, but also if specified correctly, can help to find exactly those functions and hence is less likely to overfit and underfit.

The perfect definition of γ and λ would be such that the minimum value of S is attained when the function we find approximates the minimum sufficient statistic and no further structure can be exploited, leaving γ to be the cost function over the irreducible noise. Therefore, it is important to specify S s.t. it approximates the Kolmogorov complexity of the conditional. If S gives too much weight to γ, we prioritize minimizing the error, which will lead to overfitting. On the other hand, if we define λ such that it grows too fast, we over-penalize complexity and underfit.

To illustrate this, consider Example 7.1 again. If we assign too little weight to the complexity of the function, we could probably train a deep neural network for the anti-causal direction that has a similar regression error as the simple causal model. Luckily, model selection is not a new topic and there already exist model selection criteria that try to avoid overfitting and aim at recovering the true function (Schwarz, 1978; Akaike, 1983; Grünwald, 2007). Interesting for us are only those that can be specified as an IRSF. We provide a selection of those below.

The most well-known scoring functions that we can write as an IRFS according to Definition 7.1 are the Akaike information criterion (Akaike, 1983) (AIC) and the Bayesian information criterion (Schwarz, 1978) (BIC).

Akaike Information Criterion

For the causal direction the Akaike information criterion can be written as

$$n \log(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2]) + 2\|\beta_\phi\|_0 + c\,,$$

where c is a constant term independent of the model. As the sample size n is the same for the causal and the anti-causal direction, we can consider it as a parameter of the function γ and write down an IRSF with $\gamma(a) = n \log(a)$ and $\lambda(b) = 2b + c$.

Bayesian Information Criterion

The Bayesian information criterion for scoring the causal direction is equal to

$$n \log(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2]) + \log(n) \cdot \|\beta_\phi\|_0$$

and can similar to AIC be written as an IRSF. Hence, both scores can be used in Theorem 7.1. One detail that we have to consider for AIC and BIC is that log is not defined for 0 and is negative for values between 0 and 1. Hence, it is necessary to adjust both scores by taking $\log(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2] + 1)$.

Minimum Description Length Principle

As discussed in the previous chapters, MDL codes (see Section 3.1.2) do also offer a well defined way to balance model complexity and the complexity of the data given a model (Grünwald, 2007). Defining an optimal encoding for continuous data without making any assumptions, however, is a hard problem. One approach that utilizes two-part MDL to approximate the algorithmic Markov condition for continuous data is Slope (Marx and Vreeken, 2019b), as we saw in Chapter 6. In Slope, the main assumption is that the error is Gaussian distributed. Crudely speaking, the score used in Slope can be written as $\gamma(\mathbb{E}[(\tilde{Y}_\alpha - \phi(X))^2])$, where γ is based on the negative log likelihood, plus a function ρ over the parameters. As this function ρ does not purely consider the number of parameters, but assigns different weights according to the value, of the parameter, the corresponding scoring function is not an IRSF and hence Theorem 7.1 does not apply for Slope. If we loosen the encoding from Slope slightly and "forget" about the exact values of the parameters but encode each parameter with the same constant number of bits, we arrive at an IRFS and Theorem 7.1 can be applied. In this case, the encoding would be called *lossy* as we do not encode all the information available to us.[2]

7.4 The Sloppy Algorithm

In theory, there are many possible ways to instantiate our framework, as we can use every function learning algorithm that minimizes the regression error and allows to control the number of parameters. During our empirical evaluation, we evaluate two possible ways, one using basis functions and one splines.

We refer to our method as Sloppy. We name it such both because it is partially inspired by Slope, because it is the first instantiation of the IRSF framework, but primarily because from an information-theoretic perspective the notion of a constant penalty per parameter can be inefficient (too high), as well as lossy (too low), and hence, sloppy. In practice, we consider the following two variants,

1. **Sloppy$_B$:** We find the best linear combination according to the given score function S from a set of basis functions that include polynomials up to a degree of six, an exponential and logarithmic basis function as well as reciprocal up to the degree of two. This can be done with an algorithm following the standard forward-backward selection scheme.
2. **Sloppy$_S$:** We fit a cubic spline, where we control the degrees of freedom and find that selection of splines, for which S is minimal. Even, when we do this exhaustively, Sloppy$_S$ is still very fast in practice.

[2] Such an encoding could also mean that we assume that all parameters are drawn from the same distribution and hence use a fixed amount of bits to encode them.

For our experiments, we use AIC and BIC as scoring function S.

7.4.1 Inference

Before applying Sloppy, we standardize X and Y to zero mean and unit variance or normalize them between zero and one, depending on our prior beliefs. Hence, we can assume that $K(P_X) \overset{+}{=} K(P_Y)$ and can infer the causal direction according to Theorem 7.1, as described in the previous section. Then, given an IRSF S, we use Sloppy to compute those functions ϕ and ψ that minimize $S(Y \mid X, \phi)$ and $S(X \mid Y, \psi)$. We decide that $X \rightarrow Y$ if $S(Y \mid X, \phi) < S(X \mid Y, \psi)$, that $Y \rightarrow X$ if $S(Y \mid X, \phi) > S(X \mid Y, \psi)$ and do not decide in case of equality.

7.4.2 Confidence

The authors of RECI (Blöbaum et al., 2018) showed that in empirical evaluations we can use the minimum of the error terms for both directions divided by the maximum as a confidence measure. We do so accordingly and define the confidence of a decision as

$$C(X,Y) := 1 - \frac{\min\{S(Y \mid X, \phi), S(X \mid Y, \psi)\}}{\max\{S(Y \mid X, \phi), S(X \mid Y, \psi)\}} \ .$$

The higher $C(X,Y)$, the more certain we are that our decision is correct. This allows to order decisions across different inferences by their confidence. In addition, we can set a threshold t such that we require $C(X,Y) \geq t$ and otherwise do not decide for a direction as we are not confident enough about the decision. Notably, similar to the confidence score provided in the previous chapter, $C(X,Y)$ is also a normalized score and hence is not dependent on the sample size, as we will see in the experiments.

7.5 Experiments

In this section, we empirically evaluate Sloppy and benchmark it against competing state-of-the-art methods. To represent additive noise models, we select RESIT (Peters et al., 2014) using the Hilbert Schmidt Independence Criterion to measure the independence between cause and noise distribution (Gretton et al., 2008). A recent study shows that the overall performance of RESIT on simulated and real-world data is on par if not better than competing methods of this type (Tagasovska et al., 2020). In addition, we compare against IGCI (Janzing et al., 2012) representing methods for the low-noise setup and

QCCD (Tagasovska et al., 2020), which is to the best of our knowledge the method with the best overall performance.

We first provide a detailed outline on how we configured each approach. Then, we show the overall performance over synthetic and real-world benchmark data sets, followed by a more detailed analysis. Since Sloppy$_S$ and Sloppy$_B$ performed very similar, we only present the results for Sloppy$_S$ for these experiments, which for conciseness we will refer to as Sloppy. In the last part of the experiments, in Section 7.5.6, we also provide the results for Sloppy$_B$, as well as compare to the more related regression-based methods Slope, CAM and RECI.

All experiments were performed single threaded and Sloppy took only up to a couple of seconds for a single pair. For research purposes and to make our results reproducible, we make the code for Sloppy available online.[3]

7.5.1 Configuration of Sloppy and Competing Methods

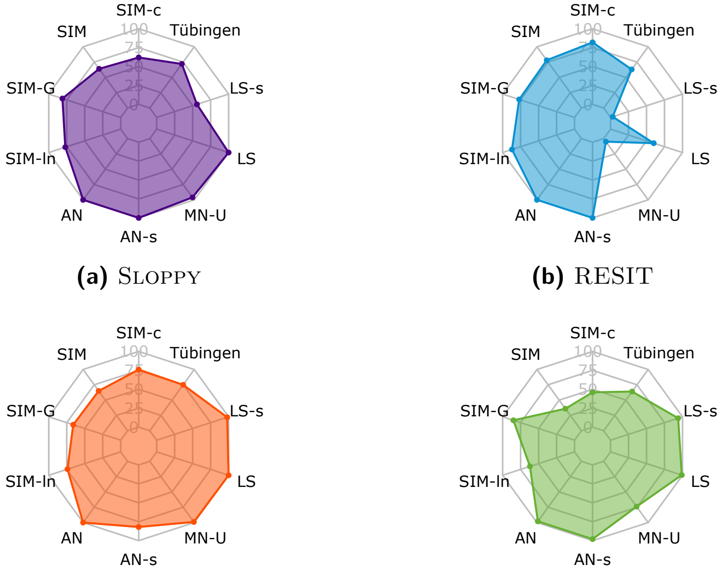
For RESIT and QCCD, we used the default configurations as recommended by the authors (Peters et al., 2014; Tagasovska et al., 2020). Before we applied IGCI to the synthetic data sets, we standardized X and Y to have zero mean and unit variance. As for all of the simulated data sets the cause was generated as a Gaussian or near Gaussian distributed random variable this preprocessing step led to better results than normalizing the data. However, when we applied IGCI to the Tübingen data set, we found that normalizing the data between zero and one led to better results, hence we reported those results.

We implemented Sloppy using cubic splines, as described in Section 7.4. Equivalent to the preprocessing we did for IGCI, we standardized X and Y to have zero mean and unit variance for the simulated data sets, as for those we knew that the cause was generated with a Gaussian or near Gaussian distribution. Since we did not know the distributions for the real-world data sets beforehand, we choose to use a uniform prior and normalized the data between zero and one for the Tübingen data set. As scoring function we used BIC for the simulated pairs. For the normalized real-world data sets, however, BIC was too restrictive and mainly fitted linear models. Hence, we used AIC for these.

7.5.2 Benchmarking

In order to benchmark Sloppy against RESIT, QCCD and IGCI, we applied them to ten benchmark data sets and reported their accuracies. We took five data sets from Mooij et al. (2016). Those consist of four simulated data sets generated using a Gaussian process: *SIM* (without confounder), *SIM-ln* (with

[3] `http://eda.mmci.uni-saarland.de/sloppy/`



(a) Sloppy

(b) RESIT

(c) QCCD

(d) IGCI

Figure 7.3: Accuracy of Sloppy, RESIT, QCCD and IGCI over all synthetic data set and the Tübingen benchmark data set.

low noise), *SIM-G* (with distributions close to Gaussian) and *SIM-c* (with confounder). The fifth one is a collection of 99 real-world bivariate continuous cause effect pairs, known as the Tübingen benchmark data set (version from December 17), for which we weigh the pairs as recommended. The remaining five data sets were taken from Tagasovska et al. (2020). These consist of non-linear functions with additive noise (*AN*), sigmoidal functions with additive noise (*AN-s*), non-linear and sigmoidal location scale functions (*LS* and *LS-s*), that is, we generate the effect Y as

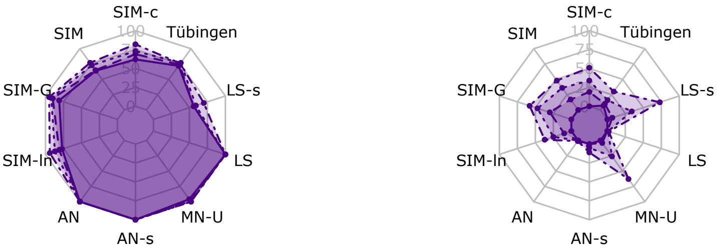
$$Y = f(X) + g(X) \cdot N_Y$$

and sigmoid functions with multiplicative uniform noise N_Y (*MN-U*)—i.e.,

$$Y = f(X) \cdot N_Y \, .$$

All simulated data sets consist of 100 cause-effect pairs. For each such pair, 1 000 samples from the joint distribution were generated.

In Figure 7.3, we show the accuracies for Sloppy, RESIT, QCCD and IGCI on all data sets. On average, Sloppy has an accuracy of 81%. If we



(a) accuracy [%]

(b) undecided [%]

Figure 7.4: Accuracy of Sloppy (left), for those decisions that have a higher confidence than $\{0, 0.01, 0.05, 0.1\}$ and right the corresponding percentage of draws.

consider only those data sets for which our assumptions hold, those are *AN* and *AN-s*, we have an accuracy of 100%. The reason why we could not achieve this for the *SIM* data sets is because they also contain pairs for which the function is close to linear, have high noise or are sampled from mixture models. Taking this under consideration, Sloppy still performs very well on these data sets. The only data set where we have a poor performance is *LS-s*, which violates our assumption with respect to the generating model. This is also the only data set where we clearly lose to QCCD. In turn, we perform better than QCCD on *AN-s* and are on par for the remaining data sets. Overall, RESIT and IGCI have more problems than Sloppy with those data sets that do not follow their assumptions and thus cover a smaller area than Sloppy.

7.5.3 Setting a Confidence Threshold

In this experiment, we consider the same data sets as above and look at the confidence of Sloppy. In particular, we show in Figure 7.4 how the accuracy of Sloppy improves when we only consider those decisions with a confidence greater than or equal to $\{0, 0.01, 0.05, 0.1\}$. We can observe that setting a threshold of 0.1 improves the average accuracy over all data sets from 81% to 89%, which clearly shows that we assign low confidence values to bad decisions. In addition, we show the percentage of pairs that do not reach the corresponding threshold. We undoubtedly see that this number is higher for those data sets that do not fulfil our assumptions, whereas for those data sets that do, the number of pairs where we do not decide remains low, even for a cut-off of 0.1.

7.5.4 Accuracy Curves

Related to confidence values are accuracy curves (see Section 6.5.1). In particular, we obtain an accuracy curve, if we order a set of decisions by their

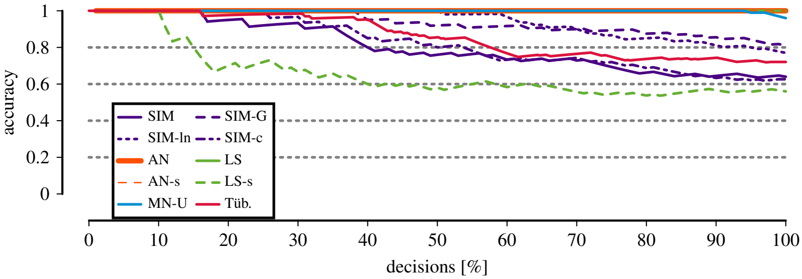


Figure 7.5: [Higher is better] Accuracy curves of Sloppy for every tested data set. As we obtain 100% accuracy for *AN*, *AN-s* and *LS*, those curves lie above each other.

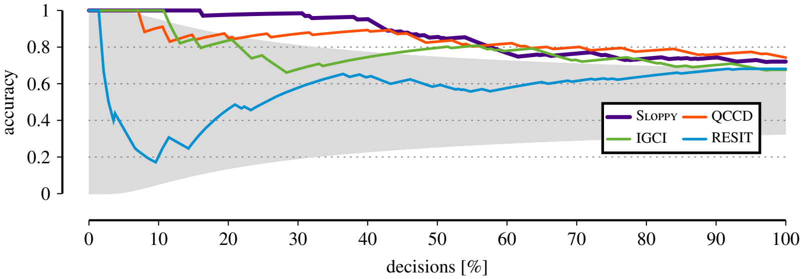
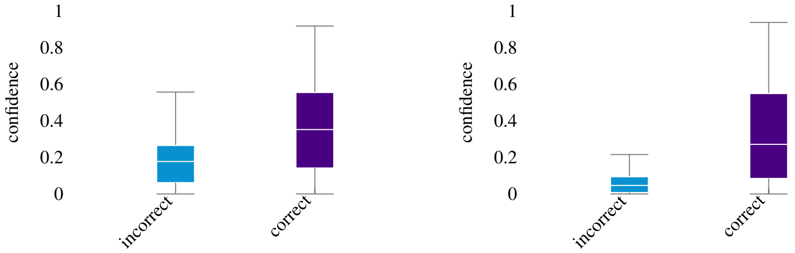


Figure 7.6: [Higher is better] Accuracy curves for Sloppy, RESIT, QCCD and IGCI on the Tübingen benchmark data set. The gray area marks the 95% confidence interval of a random coin flip.

confidence values and report for each percentage k (decision rate) the accuracy over the top k% of the decisions. In Figure 7.5 we report the accuracy curves of Sloppy for each tested data set. Importantly, we observe that for all data sets, even for *LS-s*, the first 10% of our decisions are correct. Then, depending on the overall accuracy that we achieve on the corresponding data set, the accuracy slowly drops after considering more and more decisions with lower confidence values.

In addition, we show in Figure 7.6 the accuracy curves for Sloppy, RESIT, QCCD and IGCI for the real-world benchmark data set. Although the overall performance of all methods does not differ too much, we can clearly see that Sloppy has the best accuracy curve. In particular, for the first 31% of all decisions, we only get one decision wrong and only drop below 95% accuracy after considering more than 40% of all decisions. In comparison, the competing approaches more frequently assign high confidence values to wrong decisions.



(a) Tübingen

(b) Simulated

Figure 7.7: Distribution of confidence values for correct and incorrect decisions for the Tübingen benchmark data set (left) and the simulated data sets (right).

7.5.5 Confidence Distribution

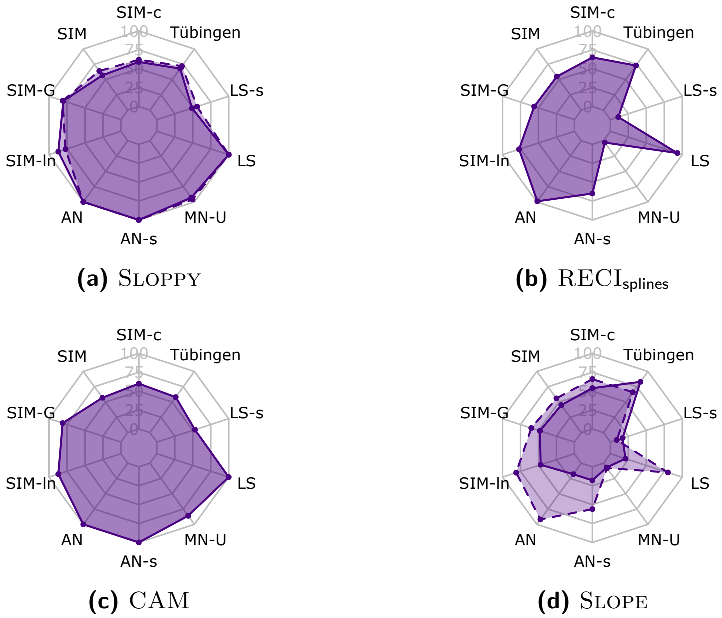
In the last experiment, we consider the distribution of the confidence values for correct and incorrect decisions for the real-world pairs and the simulated pairs, as shown in Figure 7.7. It is encouraging to see that there is a clear difference in the distribution and higher confidence values are assigned to correct decisions. For the simulated data, the first quartile for the incorrect decisions (0.094) is approximately on the same level as the third quartile for the correct decisions (0.085), which means that we could almost separate the correct form the incorrect decisions using a threshold in this region. For the real-world data the distributions overlap a bit more, however, when applying the exact implementation of the Wilcoxon-Mann-Whitney test (Wilcoxon, 1945; Marx et al., 2016), we get that the confidence values for the incorrect directions are smaller than for the correct directions with a p-value $< 10^{-4}$.

7.5.6 Comparison to RECI, CAM and Slope

In the following, we explain how we needed to configure Sloppy to obtain similar results to RECI, CAM and Slope and briefly discuss their differences. Additionally, we provide the results for Sloppy$_{\mathcal{B}}$, which turn out to be almost identical to the ones for Sloppy$_{\mathcal{S}}$.

RECI

As RECI assumes that the true functions are known, it is hard to do a fair comparison without preselecting for a suitable regressor (Blöbaum et al., 2018). To provide an impression of the results, we preprocessed the data by normalizing it between zero and one (as suggested by the authors) and then applied Sloppy$_{\mathcal{S}}$ with zero penalty for the parameters. First of all, we observe that



(a) Sloppy

(b) RECI$_{\text{splines}}$

(c) CAM

(d) Slope

Figure 7.8: Accuracy of Sloppy (solid: Sloppy$_B$, dashed: Sloppy$_S$), RECI (using cubic splines), CAM and Slope (solid: Slope allowing for non-deterministic functions, dashed: Slope using a mixture of deterministic basis functions) over all synthetic data sets and the Tübingen benchmark data set.

the splines strongly overfit, where the average number of degrees of freedom is over 140, in contrast to Sloppy, where the average number of degrees of freedom is 5. Nonetheless, the results on the synthetic and benchmark data are still reasonable, as shown in Figure 7.8. The overall average performance, however, drops from 81% to 60%. Since the authors of RECI suggest to only fit low degree polynomials (Blöbaum et al., 2018), choosing splines is, however, a sub-optimal choice.

CAM

In the bivariate setting, CAM is very related to Sloppy. CAM also uses regularized splines and standardizes the data, where they maximize the log-likelihood for both directions (Bühlmann et al., 2014). Therefore, it is not surprising, that the results obtained with CAM are very similar to the results we get with Sloppy, as shown in Figure 7.8. However, when CAM was developed, the authors only showed consistency of their method for finding a hierarchy among a set of variables in a causal discovery setup and did not focus

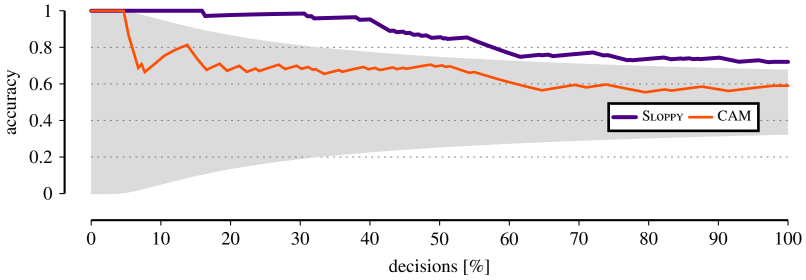


Figure 7.9: Decision rates for Sloppy and CAM on the Tübingen benchmark.

on the bi-variate case. In this chapter, we focused especially on identifiability in the bi-variate setting. In addition, when we compare the decision rates of CAM and Sloppy on the Tübingen benchmark data set (see Figure 7.9), we see that Sloppy clearly outperforms CAM.

Slope

For Slope, we also standardize the data between zero and one. In particular, there exist two versions: Slope using a deterministic function and allowing for non-deterministic functions (Marx and Vreeken, 2017) and Sloper using a set of basis functions, without fitting non-deterministic functions (Marx and Vreeken, 2019b). Apart from the exact score and the preprocessing, Sloper comes close to Sloppy$_B$. When we look at the results over all data sets (Figure 7.8), we see that Sloper performs similar to RECI using cubic splines. On average, Slope performs much worse than Sloper and only has a better performance on the Tübingen benchmark data set.

7.6 Conclusion

We considered causal inference between two continuous random variables X and Y, without hidden confounders. The main contribution in this chapter is that we showed under which conditions we can use regularized regression to identify cause from effect with guarantees. Further, we linked this result to Slope, which we proposed in the previous chapter.

As a possible instantiation of our framework, we introduced Sloppy—which finds the best fitting function for the causal and anti-causal direction according to a given identifiable regression-based scoring function. In practice, we model functions using either a set of basis functions or cubic splines and use AIC or BIC as scoring functions. Our results show that Sloppy outperforms the state-of-the-art algorithms with identifiability guarantees on synthetic and real-world data and is on par with methods that do not have such guarantees. We note, however, that SLOPPY is just a first instantiation and are quite certain it is possible to define—and are looking forward to seeing—instantiations of IRFS that will outperform our method in practice.

In the next chapter, we will take a step back and look at a more general approach that considers causal inference between two multivariate mixed random variables. This approach, similar to the work presented in the previous chapter, builds upon the two-part MDL approximation of the algorithmic independence of conditionals postulate.

Chapter 8

Causal Inference via Classification and Regression Trees

We already saw how to make cause-effect inference between two univariate random variables of the same type. In practice, however, X and Y do not have to be of the same type. The altitude of a location (continuous), for example, determines whether it is a suitable habitat (binary) for a mountain hare. In fact, neither X nor Y has to be univariate. Whether or not a location is a good habitat for an animal is not just caused by a single aspect but by a *combination* of conditions such as altitude, annual average temperature, and precipitation, which do not have to be of the same type. Therefore, we are interested in the general case where X and Y may be of any cardinality and may be single or mixed-type. A different interpretation would be to think of X and Y as meta-variables; e.g. altitude, annual average temperature and precipitation could be summarized by a meta-variable labelled as environmental conditions.

Such a general setting has not been considered previously and we were the first to tackle this problem. As we saw in the previous chapters, causal inference for two univariate random variables is a well-researched topic (see Section 6.4). When it comes to multivariate variables, there exist only a few approaches. For purely continuous multivariate data, Janzing et al. (2010) proposed the linear trace method (LTR), which aims to find a structure matrix A and the covariance matrix Σ_X to express Y as AX. Under the quite restrictive assumptions

This chapter is based on Marx and Vreeken (2018).

that the functions are deterministic and invertible, Chen et al. (2013) developed a kernelized version of LTR. Most related to our approach are Origo (Budhathoki and Vreeken, 2017b) and Ergo (Vreeken, 2015). Origo was developed for multivariate binary data and uses a compression-based approach to approximate the algorithmic independence of conditionals. To instantiate their approach in practice, they use binary trees. Ergo, on the other hand, is based on the postulate that the cause contains more relative information about the effect than vice versa—i.e., if $X \rightarrow Y$, $K(Y|X)/K(Y) < K(X|Y)/K(X)$ (Vreeken, 2015). The author instantiates this framework for multivariate continuous data by using cumulative entropies.

In this work, we consider multivariate and mixed-type data. In particular, we define an MDL score for coding forests, a model class where a model consists of classification and regression trees, as this allows us to consider both discrete and continuous-valued data with one unified model. By allowing dependencies from X to Y, or vice versa, we can also measure conditional complexities and hence use our encoding for causal inference. We first briefly discuss the two different causal indicators, the one based on the algorithmic independence of conditionals and the one developed for Ergo in Section 8.1. In addition, we develop an adjusted version of the Ergo indicator, which adjusts for biases induced by mixed variables and large differences in the domain space of the involved variables. After that, in Section 8.2, we discuss how to encode classification and regression forests. To learn such forests from data, we present the Crack algorithm, short for **c**lassification and **r**egression-based p**ack**ing, in Section 8.3. Last, we empirically evaluate both causal indicators in Section 8.4. We first evaluate the differences between both variants and then compare them on both univariate benchmark data and multivariate pairs of different types.

8.1 Causal Inference by Compression

As in Chapters 5 and 6, we discuss the algorithmic independence of conditionals or, more specifically, its two-part MDL approximation. Chapter 5 discussed general aspects about the connection between two-part MDL and AIC and Chapter 6 focused on a specific instantiation of it for univariate numeric data. Here, we develop a two-part MDL approach where X and Y can be multivariate and of mixed type, i.e., binary, or generally discrete, or continuous.

In the following, we are going to very briefly recap the algorithmic independence of conditionals for two random variables and its corresponding two-part MDL approximation. Then we discuss the challenges that occur when considering multivariate mixed random variables and propose an alternative causal indicator based on two-part MDL.

8.1.1 Causal Inference by Complexity

As described in more detail in Section 5.2, the general idea can be summarized as follows. Connected to the idea of independent mechanisms, Janzing and Schölkopf (2010) postulated the algorithmic independence of conditionals, which for two variables states that if $X \to Y$,

$$K(P_X) + K(P_{Y|X}) \leq K(P_Y) + K(P_{X|Y}) \; .$$

To approximate the above inequality using two-part MDL for data (x^n, y^n), we need to define a model class. For the causal direction, we define a model as $M_{X\to Y} = (M_X, M_{Y|X})$ from the class $\mathcal{M}_{X\to Y} = \mathcal{M}_X \times \mathcal{M}_{Y|X}$ that best describes the data over Y by exploiting as much structure of X as possible to save bits. By MDL, we identify the optimal model $M_{X\to Y} \in \mathcal{M}_{X\to Y}$ for data over X and Y as the one minimizing

$$L_{X\to Y} := L(M_X) + L(x^n \mid M_X) + L(M_{Y|X}) + L(y^n \mid x^n, M_{Y|X}) \; ,$$

where we encode a model for the anti-causal direction equivalently. In the following, we will refer to the above, i.e., $L_{X\to Y}$ and its analogue for the inverse direction $L_{Y\to X}$, as the *Absolute Causal Indicator* (ACI). We will refer to $L_{X\to Y}$ as $ACI_{X\to Y}$ and to $L_{Y\to X}$ as $ACI_{Y\to X}$. Akin to the Kolmogorov complexity based inference criterium, we infer that X is a likely cause of Y if $ACI_{X\to Y} < ACI_{Y\to X}$, Y is a likely cause of X if $ACI_{X\to Y} > ACI_{Y\to X}$ and do not decide between both alternatives if $ACI_{X\to Y} = ACI_{Y\to X}$.

8.1.2 Normalized Causal Indicator

The absolute causal indicator has nice theoretical properties that follow from the algorithmic independence of conditionals. However, by considering the absolute difference in encoded lengths between $X \to Y$ and $Y \to X$, it has an intrinsic bias towards data of higher marginal complexity. For example, when we gain 5 bits between encoding the data over Y conditioned on X, rather than independently, this is more impressive if $L(y^n \mid M_Y) \approx 100$ as opposed to $L(y^n \mid M_Y) \approx 1\,000\,000$ bits. This is particularly important in the mixed-data setting, as the marginal complexity of a binary attribute will typically be much smaller than that of an attribute recorded at a higher resolution.

To address this shortcoming of ACI, we propose a novel, normalized indicator for causal inference on mixed-type data. We start with the Ergo indicator (Vreeken, 2015), which rather than the absolute difference considers the compression *ratios* of the target variables, i.e., if $X \to Y$ then

$$\frac{L(M_{Y|X}) + L(y^n \mid x^n, M_{Y|X})}{L(M_Y) + L(y^n \mid M_Y)} < \frac{L(M_{X|Y}) + L(x^n \mid y^n, M_{X|Y})}{L(M_X) + L(x^n \mid M_X)} \; .$$

That is, the fraction of bits we save when encoding the effect conditioned on the cause relative to encoding only the effect is higher than for the anti-causal direction, i.e., conditioning the cause on the effect. This score accounts for different marginal complexities of X and Y, and hence suffices for the univariate mixed-type data case. For multivariate and mixed-type data, we face the same problem: if the variables according to individual dimensions $Y_i \in Y$ are of different marginal complexities $L(y_i^n \mid M_{Y_i})$, the gain in compression of one single Y_i may dominate the overall score simply because it has a larger marginal complexity (e.g. because it has a large sample space).

We can compensate this by explicitly considering the compression ratios *per dimension* $Y_i \in Y$, rather than the compression ratio over Y as a whole. Formally, we define our new *Normalized Causal Indicator* (*NCI*) as

$$NCI_{X \to Y} = \frac{1}{|Y|} \sum_{Y_i \in Y} \frac{L(y_i^n \mid x^n, M_{Y_i|X}) + L(M_{Y_i|X})}{L(y_i^n \mid M_{Y_i}) + L(M_{Y_i})} .$$

As above, we infer $X \to Y$ if $NCI_{X \to Y} < NCI_{Y \to X}$, infer $Y \to X$ if $NCI_{X \to Y} > NCI_{Y \to X}$ and do not decide if both scores are equal.

Although free of bias from the marginal complexities of individual variates, we have to be careful to screen for redundancy within Y resp. X. By definition, *NCI* counts the causal effect on each variate, and redundancies within Y (resp. X) hence exacerbate the measured effect. This is, however, not problematic as we can detect such redundancies using standard independence tests.

In practice, we expect that *ACI* performs well on data where X and Y are of the same type, especially when $|X| = |Y|$ and the domain sizes of their attributes are balanced. Whenever the variates of X and Y are of different marginal complexities, e.g., because of unbalanced domains, dimensionality, and especially for mixed-type data, the experiments confirm that the *NCI* performs much better than the *ACI*.

8.2 MDL for Tree Models

To use MDL in practice, we need to specify an appropriate model class $\mathcal{M}$, and define how to encode both data and models in bits. Here, we need to be able to consider continuous (up to a certain precision), discrete and mixed-type data, be able to exploit dependencies between attributes of different types, and be able to encode the data of Y conditioned on the data of X. Classification and regression trees (CART) lend themselves very naturally to do all of this.

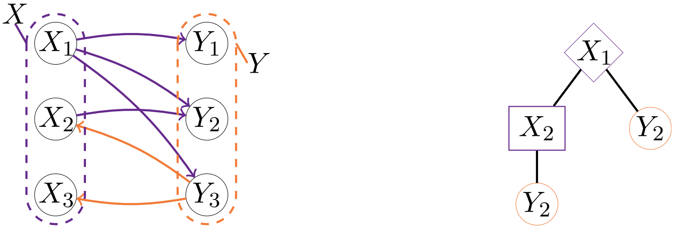
To formally write down our MDL model, we will use the following notation. We consider two multivariate mixed-type random variables X and Y and further consider $\boldsymbol{A} = X \cup Y$, i.e., the set containing both X and Y with $|\boldsymbol{A}| = m$ dimensions. In the following, we will often refer to a single dimension or univariate random variable $A_i \in \boldsymbol{A}$ as an attribute, regardless of whether it belongs to X or Y. An attribute A_i has a type, where $\text{type}(A_i) \in \{\textit{binary}, \textit{categorical}, \textit{continuous}\}$. We will refer to binary and categorical attributes as *nominal* attributes. We write $\mathcal{A}_i$ to denote the domain of an attribute A_i. Respectively, the size of the domain of an attribute $|\mathcal{A}_i|$ is for discrete data simply the number of distinct values and for numeric data equal to $\frac{\max(A_i)-\min(A_i)}{\tau_{A_i}} + 1$, where τ_{A_i} is the resolution at which the data over attribute A_i was recorded. For example, a resolution of 1 means that we consider integers, of 0.01 means that the corresponding attribute was recorded with a precision of up to a hundredth.

8.2.1 Encoding a Tree

In the following, we discuss how we can model a set of attributes $\boldsymbol{A}$ and dependencies among them using tree models. That is, we consider models M that contain a classification or regression tree T_i per attribute $A_i \in \boldsymbol{A}$, where tree T_i encodes the data over A_i by exploiting dependencies on other attributes by means of splitting or regression. In particular, an internal node $v \in \text{int}(T_i)$ of a tree T_i models a dependency to another attribute, represented by a split or a regression step—i.e., internal nodes relate to the model. For example, a tree T_i that contains a split node w.r.t. a binary attribute A_j and a regression node w.r.t. attribute A_k models the data of attribute A_i conditioned on $\{A_j, A_k\}$. To encode the data, we consider the leave nodes $\text{lvs}(T_i)$ of a tree T_i. A leaf node $l \in \text{lvs}(T_i)$ describes the data with respect to the number of data points $|l|$ associated to the corresponding leaf. Loosely speaking, the better we can fit the data by creating dependencies on other nodes, the more succinctly we will be able to encode the data in the leave nodes.

To ensure lossless decoding, there needs to exist an order on the trees $T_i \in M$ such that we can transmit these one by one. In other words, in a *valid* tree model there are no cyclic dependencies between the trees $T_i \in M$, and a valid model can hence be represented by a DAG. Let $\mathcal{M}_{\boldsymbol{A}}$ be the set of all valid tree models for $\boldsymbol{A}$, that is, $M \in \mathcal{M}_{\boldsymbol{A}}$ is a set of $|\boldsymbol{A}|$ trees such that the data types of the leafs in T_i corresponds to the data type of attribute a_i, and its dependency graph is acyclic.

We write $\mathcal{M}_X$ and $\mathcal{M}_Y$ to denote the subset of valid coding forests for X and Y, where we do not allow dependencies. To describe the possible set of models where we allow attributes of X to only depend on attributes of Y we write $\mathcal{M}_{X|Y}$ and do so accordingly for Y depending only on X. If an attribute does not have any incoming dependencies, its tree is a stump. Figure 8.1 shows the DAG for a toy data set, and an example tree for Y_2. From the DAG, the set of purple edges would be a valid model in $\mathcal{M}_{Y|X}$, whereas the orange edges are a valid model for $\mathcal{M}_{X|Y}$.



(a) DAG

(b) Tree for Y_2

Figure 8.1: Toy data set with ground truth $X \to Y$. Shown is the dependency DAG (right). More dependencies go from X to Y than vice versa. Left: Example coding tree for Y_2. X_1 splits the values of Y_2 into two subsets. In addition, the subset belonging to the left child can be further compressed by regressing on X_2.

Now that we know the relevant model classes, we can define our MDL score. At the highest level, the number of bits to describe data $\boldsymbol{a}^n$ over attributes $\boldsymbol{A}$ together with a valid model M for $\boldsymbol{A}$ as

$$L(\boldsymbol{a}^n, M) = \sum_{T_i \in M} L(a_i^n, T_i) \ ,$$

where we make use of the fact that M encodes a DAG structure, and we can hence serialize its dependencies.

In turn, the encoded cost of a tree T_i consists of two independent parts. First, we transmit its topology, and second the data in its leaves. For the topology, we indicate per node whether it is a leaf or an internal node, and if the latter, whether it is a split or regression node. Formally we hence have

$$L(a_i^n, T_i) = |T_i| + \sum_{v \in \operatorname{int}(T_i)} (1 + L(v)) + \sum_{l \in \operatorname{lvs}(T_i)} L(l) \ .$$

This leaves us to define the encoded cost of an internal node, $L(v)$, and the encoded cost of the data in the leaves, $L(l)$.

Cost of a Node

A node $v \in T_i$ can be of two main types; it either defines a split, or a regression step. We consider these in turn. With regard to splits, we include both multiway and binary splits. To encode a split node v, we need

$$L_{\text{split}}(v) = 1 + \log |\boldsymbol{A}| + L(\Phi_{\text{split}})$$

bits. We first encode whether it is a single or multiway split, then the attribute A_j on which we split, and last the conditions on which we split. For single way splits, $L(\Phi_{\text{split}})$ corresponds to the cost of describing the value in the domain of X_j on which we split, which is equal to $\log|\mathcal{A}_j|$ when A_j is categorical, and $\log(|\mathcal{A}_j|-1)$ when it is binary or numeric. Note that we consider these two cases for categorical and binary data, since for binary data there is only a single option for a split, whereas for categorical data any value in $\mathcal{A}$ can be the one we split on. For multiway splits on categorical attributes A_j we split on all values, which costs no further bits, while for numeric A_j we split on every value that occurs at least k times—with one residual split for all remaining data points. To encode k we use $L_{\mathbb{N}}$, the MDL optimal code for integers (Rissanen, 1983).

To encode a regression node v, we first encode the attribute we regress on, and then the parameters $\Phi(v)$ of the regression function, i.e.,

$$L_{\text{reg}}(v) = \log|\boldsymbol{A}| + \sum_{\phi\in\Phi(v)} \left(1 + L_{\mathbb{N}}(s) + L_{\mathbb{N}}(\lceil|\phi|\cdot 10^{s}\rceil)\right).$$

Similar as for Slope, we encode each parameter $\phi \in \Phi(v)$ up to user defined number of significant digits s. In practice, for computational reasons, we use linear and quadratic regression, but note that this score is general and can encode any regression function with a real-valued parameter vector.

Cost of a Leaf

In classification and regression trees, the actual data is stored in the leaves. To encode the data in a leaf of a categorical or discrete attribute, we can assume a multinomial distribution and encode the data using the normalized maximum likelihood (Kontkanen and Myllymäki, 2007), which we defined in Section 3.2. In particular, we encode the data of a categorical leaf using the stochastic complexity for multinomials as

$$L_{\text{nom}}(l) = |l|\cdot\hat{H}(A_i \mid l) + \log\mathcal{C}^{|l|}_{|\mathcal{A}_i|}\,,$$

where $\hat{H}$ denotes the empirical Shannon entropy, $|\mathcal{A}_i|$ domain size of attribute A_i and $|l|$ is the number of data points associated with the current leaf.

For numeric data, refined MDL encodings have very high computational complexity (Kontkanen and Myllymäki, 2007). In the interest of efficiency, we hence encode the data in numeric leaves with two-part MDL, using point models with a Gaussian, resp. uniform distribution. The former is especially suited for encoding residuals, since such a step aims to minimizes the variance of Gaussian distributed error. A split or a regression node can reduce the variance, or the domain size of the data in the leaf, and each can therewith reduce the cost. The costs for a leaf assuming a Gaussian distribution are

$$L_{\text{num}}(l \mid \hat{\sigma}, \hat{\mu}) = \frac{|l|}{2} \left(\frac{1}{\ln 2} + \log 2\pi\hat{\sigma}^2 \right) - |l| \log \tau_{A_i} ,$$

given empirical mean $\hat{\mu}$ and variance $\hat{\sigma}$ or as uniform given min and max

$$L_{\text{num}}(l \mid \min(l), \max(l)) = |l| \cdot \log \left(\frac{\max(l) - \min(l)}{\tau_{A_i}} + 1 \right) .$$

We encode the data as Gaussian if this costs fewer bits than encoding it as uniform. To indicate this decision, we use one bit and encode the minimum of both plus the corresponding parameters. As we consider empirical data, we can safely assume that all parameters fall within the domain of the given attribute. The encoded costs of a numeric leaf l hence are

$$L_{\text{num}}(l) = 1 + 2 \log |\mathcal{A}_i| + \min\{L_{\text{num}}(l \mid \hat{\sigma}, \hat{\mu}), L_{\text{num}}(l \mid \min(l), \max(l))\} .$$

We now have a complete score. In the next section we discuss how to optimize it, but first we discuss some important causal aspects.

Identifiability and Limitations

Tree models are closely related to the algorithmic model of causality as postulated by Janzing and Schölkopf (2010). That is, every node x_i in a DAG can be computed by a program q_i with length $O(1)$ from its parents pa_i and additional input n_i—formally, $x_i = q_i(pa_i, n_i)$. Following the AIC postulate, the shortest description of x_i is through its parents.

In general, the MDL optimal tree model identifies the shortest description of a node A_i conditioned on a subset of attributes $\boldsymbol{S}_i \subseteq \boldsymbol{A} \backslash \{A_i\}$. In particular, by splitting or regressing on an attribute $A_j \in \boldsymbol{S}_i$ it approximates program q_i given the parents as input. The remaining unexplained data that corresponds to the additional input or noise n_i is encoded in the leaves of the tree. In other words, tree T_i with the minimal costs relates to the tree where $\boldsymbol{S}_i$ contains only the parents of A_i, and encodes exactly the relevant dependencies towards A_i.

Although tree models are very general, we can identify specific settings in which the model is identifiable. First, consider the case where X and Y are univariate and of a single type. If both are continuous, our model reduces to a simple regression model. If the complexities of the marginal codes are equal, we can build upon the identifiability results for almost deterministic, non-linear functions developed for regularized regression (Blöbaum et al., 2018; Marx and Vreeken, 2019c). Similarly, building upon the algorithmic independence of conditionals, for discrete data we can identify additive noise models using stochastic complexity (Budhathoki and Vreeken, 2017a). Combining these re-

Algorithm 8.1: Crack($\boldsymbol{A}, \mathcal{M}$)

```
input : data $\boldsymbol{a}^n$ over attributes $\boldsymbol{A}$, model class $\mathcal{M}$
output: tree model $M \in \mathcal{M}$ with low $L(\boldsymbol{a}^n, M)$
$T_i \leftarrow$ TrivialTree($A_i$) for all $A_i \in \boldsymbol{A}$;
$\mathcal{G} \leftarrow (V = \{v_i \mid A_i \in \boldsymbol{A}\}, E = \emptyset)$;
while $L(\boldsymbol{a}^n, M)$ decreases do
    for $A_i \in \boldsymbol{A}$ do
        $O_i \leftarrow T_i$;
        for $l \in \mathrm{lvs}(T_i), (i, j) \in \mathcal{G}$ do
            if $E \cup (v_i, v_j)$ is acyclic and $j \notin \mathrm{path}(l)$ then
                $T_i' \leftarrow$ RefineLeaf($T_i, l, j$);
                if $L(T_i') < L(O_i)$ then
                    $O_i \leftarrow T_i'$;
    $k \leftarrow \arg\min_i\{L(O_i) - L(T_i)\}$;
    if $L(O_k) < L(T_k)$ then
        $T_k \leftarrow O_k$, $E \leftarrow E \cup (v_k, v_{e_k})$ ;
return $M \leftarrow \bigcup_i T_i$
```

sults to multivariate mixed-type data is, however, non-trivial. Thus, we leave this part for future work.

In practice, we are limited by the optimality of our approximation of Kolmogorov complexity. That is, any inferences we make are with respect to the encoding we defined above, rather than the much more generally defined Kolmogorov complexity. If the generating process does not use tree-models, or measures complexity differently, the inferences we draw based on our score may be wrong. The experiments show, however, that our scores are very reliable even in adversarial settings.

8.3 The Crack Algorithm

Finding the optimal decision tree for a single nominal attribute is NP-hard, and hence so is the optimization problem at hand. We introduce the Crack algorithm, which stands for **c**lassification and **r**egression based p**ack**ing of data. Crack is an efficient greedy heuristic for discovering a coding forest M from model class $\mathcal{M}$ for data over attributes $\mathbf{A}$ with low $L(\boldsymbol{a}^n, M)$. It builds upon the well-known ID3 algorithm (Quinlan, 1986).

Greedy Algorithm

We give the pseudocode of Crack as Algorithm 8.1. Before running the algorithm, we set the resolution per attribute. To be robust to noise, we set τ_{A_i} for continuous attributes to the k^{th} smallest distance between two adjacent values, with $k = 0.1 \cdot n$.

Crack starts with an empty model consisting of only trivial trees, i.e., leaf nodes containing all records, per attribute (line 1). We iteratively discover that refinement of the current model that maximizes compression. To find the best refinement, we consider every attribute (line 4), and every legal additional split or regression of its corresponding tree (line 8). That is, a refinement is only legal when the dependency is allowed by the model family $\mathcal{M}$ (lines 6–7) and the dependency graph remains acyclic.

The key subroutine of Crack is RefineLeaf, in which we discover the optimal refinement of a leaf l in tree T_i. That is, it finds the optimal split of l over all candidate attributes A_j such that we minimize the encoded length. In case both A_i and A_j are numeric, RefineLeaf also considers the best linear and quadratic regression and decides for the variant with the best compression—choosing to split in case of a tie. In the interest of efficiency, we do not allow splitting or regressing multiple times on the same candidate.

Since we use a greedy heuristic to construct the coding trees, we have a worst case runtime of $O(2^m n)$, where m is the number of attributes and n is the number of data points. In practice, Crack takes only a few seconds for all tested cause-effect pairs.

Causal Inference with Crack

To compute our causal indicators we run Crack twice on the given data set D. First with model class $\mathcal{M}_{X|Y}$ to obtain $M_{X|Y}$ and second with $\mathcal{M}_{Y|X}$, to obtain $M_{Y|X}$. For the marginal $L(x^n \mid M_X)$ we assume a uniform prior and define $L(x^n \mid M_X) = \sum_{A_i \in X} (n \log |\mathcal{A}_i|)$ if A_i is a numeric attribute and simply take the score of a tree without any splits for a discrete variable. We encode the marginal costs for Y accordingly. We refer to Crack using *NCI* as Crack$_N$, and as Crack$_A$ using *ACI*.

8.4 Experiments

In this section, we evaluate Crack empirically. We implemented Crack in C++, and provide the source code including the synthetic data generator along with the tested datasets for research purposes.[1] The experiments concerning Crack were executed single-threaded on a MacBook Pro with 2.6 GHz Intel

[1] `http://eda.mmci.uni-saarland.de/crack/`

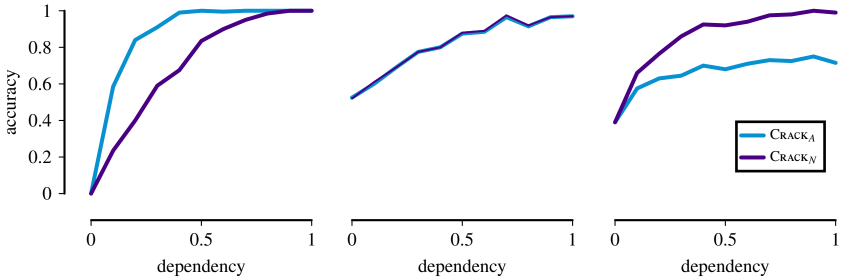


Figure 8.2: Accuracy for *ACI* and *NCI* on discrete (left), continuous (middle) and mixed-type (right) data based on dependence probability φ.

Core i7 processor and 16 GB memory running Mac OS X. All tested data sets could be processed within seconds; with a maximum runtime of 3.8 seconds.

8.4.1 Synthetic Data

On synthetic data, we want to show the advantages of either score. In particular, we expect Crack$_A$ to perform well on categorical data and continuous data with balanced domain sizes and dimensions, whereas we expect Crack$_N$ to outperform Crack$_A$ on continuous data with varying domain sizes, dimensions and on mixed-type data.

We generate synthetic data with assumed ground truth $X \rightarrow Y$ with $|X| = k$ and $|Y| = l$, each having $n = 5\,000$ rows, as follows. First, we randomly assign the type for each attribute in X. For discrete data, we randomly draw the number of classes between two (binary) and five and distribute the classes uniformly. Continuous data is generated following a normal distribution taken to the power of q by keeping the sign, leading to a sub-Gaussian ($q < 1.0$) or super-Gaussian ($q > 1.0$) distribution.[2]

To create data with the true causal direction $X \rightarrow Y$, we introduce dependencies from X to Y, where we distinguish between splits and refinements. We call the probability threshold to create a dependency $\varphi \in [0, 1]$. For each $j \in \{1, \ldots, l\}$, we throw a biased coin based on φ for each $X_i \in X$ that determines if we model a dependency from X_i to Y_j. A split means that we find a category (discrete) or a split-point (continuous) on X_i to split Y_j into two groups, for which we model its distribution independently. As refinement, we either do a multiway split or model Y_j as a linear or quadratic function of X_i plus independent Gaussian noise. We decide uniformly at random whether to do a split or refinement.

[2] To ensure identifiability, we use super- and sub-Gaussians (Hoyer et al., 2009).

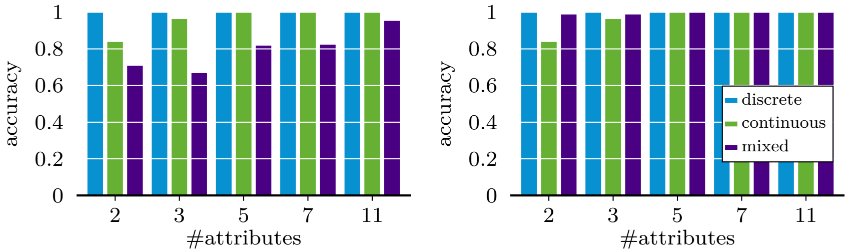


Figure 8.3: Accuracy of ACI (left) and NCI (right) on symmetric dimensions $k \in \{2, 3, 5, 7, 11\}$ for discrete, continuous and mixed-type data.

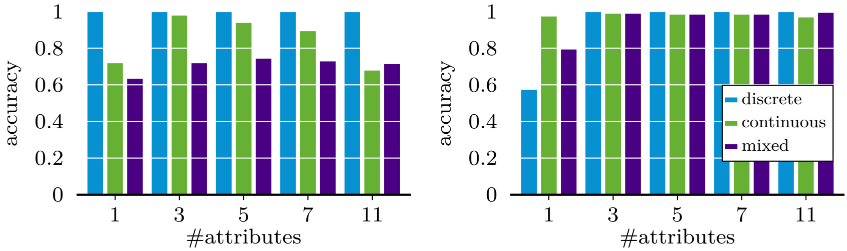


Figure 8.4: Accuracy of ACI (left) and NCI (right) on synthetically generated causal pairs of asymmetric cardinality, $|X| = 3$ and $|Y| \in \{1, 3, 5, 7, 11\}$ with ground truth $X \to Y$ or $Y \to X$ chosen randomly, for discrete, continuous and mixed-type data.

Accuracy

First, we compare the accuracies of Crack$_N$ and Crack$_A$ with regard to single-type and mixed-type data. To do so, we generate 200 synthetic data sets with $|X| = |Y| = 3$ for each dependency level where $\varphi \in \{0.0, 0.1, \ldots 1.0\}$. Figure 8.2 shows the results for discrete, continuous and mixed-type data. For single-type data, the accuracy of both methods increases with the dependency, and reaches nearly 100% for $\varphi = 1.0$. At $\varphi = 0$, both approaches correctly do not decide instead of taking wrong decisions. As expected Crack$_N$ strongly outperforms Crack$_A$ on mixed-type data, reaching near 100% accuracy, whereas Crack$_A$ reaches only 72%. On discrete data, Crack$_A$ picks up the correct signal faster than Crack$_N$.

Dimensionality

Next, we evaluate how sensitive both scores are w.r.t. the dimensionality of both X and Y, where we separately consider the cases of symmetric $k = l$

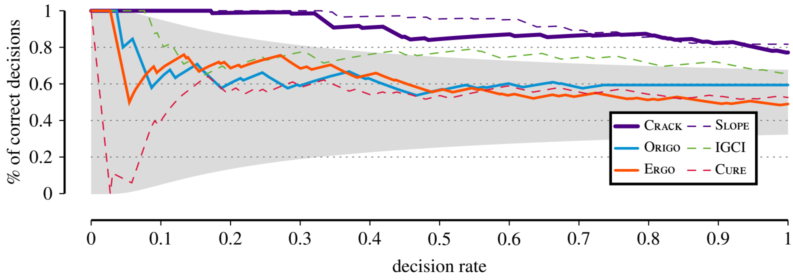


Figure 8.5: [Higher is better] Accuracy curves of the multivariate methods CRACK, ORIGO and ERGO, and the univariate methods IGCI, CURE and SLOPE (dashed lines) on the univariate Tübingen causal benchmark pairs (100), weighted as defined.

and asymmetric $k \neq l$ dimensionalities. Per setting, we consider the average accuracy over 200 independently generated data sets.

For the symmetric case, both methods are near to 100% on single-type data, whereas only CRACK$_N$ also reaches this target on mixed-type data, as can be seen in Figure 8.3. To test asymmetric pairs, we set the dimensionality of X to three, $|X| = 3$, and vary the dimensionality of Y from 1 to 11. To avoid bias, we choose the ground truth causal direction, i.e., $X \to Y$ and $Y \to X$, uniformly at random. We plot the results in Figure 8.4. We observe that CRACK$_N$ has much less difficulty with the asymmetric data sets than CRACK$_A$. CRACK$_N$ performs near perfect and has a clear advantage over CRACK$_A$ on mixed-type and continuous data. CRACK$_A$ performs better on discrete data for $l = 1$.

8.4.2 Univariate Benchmark Data

To evaluate CRACK on univariate data, we apply it to the well-known Tübingen benchmark data set (v1.0) consisting of 100 univariate cause-effect pairs with known ground truth.[3] As these are mainly numeric pairs, with only a few categoric instances, we apply CRACK$_A$. We compare to the state-of-the-art methods for multivariate pairs, ORIGO (Budhathoki and Vreeken, 2017b) and ERGO (Vreeken, 2015), and those specialized for univariate pairs, which are CURE (Sgouritsa et al., 2015), IGCI (Janzing et al., 2012) and SLOPE (Marx and Vreeken, 2017) using their publicly available implementations and recommended parameter settings.[4]

For each approach, we sort the results by confidence. Accordingly, we calculate the accuracy curves (see Section 6.5.1). We plot the results in Figure 8.5 and show the 95% confidence interval of a fair coin flip as a grey area. Except

[3] https://webdav.tuebingen.mpg.de/cause-effect/

[4] Note that SLOPPY and QCCD were published after CRACK.

Table 8.1: Comparison of LTR, Ergo, Origo and Crack on 17 multivariate cause-effect pairs with known ground truth. The type is either "N" for numeric or "M" for mixed. A "✓" indicates a correct decision, a "–" an incorrect one and (n/a) that a method is not applicable.

						Decisions			
Causal Pair	n	$\lvert X\rvert$	$\lvert Y\rvert$	Ground Truth	Type	LTR	Ergo	Origo	Crack
Climate	10 226	4	4	$Y \rightarrow X$	N	✓	✓	–	–
Ozone	989	1	3	$Y \rightarrow X$	N	(n/a)	✓	✓	✓
Car	392	3	2	$X \rightarrow Y$	N	–	✓	✓	✓
Radiation	72	16	16	$Y \rightarrow X$	N	–	–	–	✓
Symptoms	120	6	2	$X \rightarrow Y$	M	✓	✓	–	✓
Brightness	1 000	9	1	$X \rightarrow Y$	N	(n/a)	(n/a)	–	✓
Chemnitz	1 440	3	7	$X \rightarrow Y$	N	✓	✓	✓	✓
Precipitation	4 748	3	12	$X \rightarrow Y$	N	✓	–	–	✓
Stock 7	2 394	4	3	$X \rightarrow Y$	N	–	✓	–	✓
Stock 9	2 394	4	5	$X \rightarrow Y$	N	–	✓	–	✓
Haberman	306	3	1	$X \rightarrow Y$	M	✓	✓	–	–
Iris flower	150	4	1	$X \rightarrow Y$	M	(n/a)	(n/a)	–	✓
Canis	2 183	4	2	$X \rightarrow Y$	M	(n/a)	(n/a)	✓	✓
Lepus	2 183	4	3	$X \rightarrow Y$	M	(n/a)	(n/a)	✓	✓
Martes	2 183	4	2	$X \rightarrow Y$	M	(n/a)	(n/a)	✓	✓
Mammals	2 183	4	7	$X \rightarrow Y$	M	(n/a)	(n/a)	✓	✓
Octet	82	1	10	$Y \rightarrow X$	N	(n/a)	✓	✓	✓
Accuracy						0.56	0.82	0.47	0.88

to Crack none of the multivariate methods is significant w.r.t. the fair coin flip. In particular, Crack has an accuracy of over 90% for the first 41% of its decisions and reaches 77.2% overall—the final result of Crack$_N$ is only 3% worse. Crack also beats both Cure (52.5%) and IGCI (66.2%), which are methods specialized for univariate pairs. Perhaps most impressively, Crack performs within the 95% confidence interval of the best performing method, Slope, which has an overall accuracy of 81.7%. Slope is at the advantage for univariate pairs as it can exploit non-deterministic structure in the data, which is not doable for multivariate data.

8.4.3 Real World Data

Next, we apply Crack$_N$ on multivariate mixed-type and single-type data, where we collected 17 cause-effect pairs with known ground truth. We provide basic statistics for each pair in Table 8.1. The first six are part of the Tübingen benchmark (Mooij et al., 2016), and the next four were provided by Janzing et al. (2010). Further, we extracted cause-effect pairs from the *Haberman* and *Iris* (Dheeru and Karra Taniskidou, 2017), *Mammals* (Heikinheimo et al., 2007) and *Octet* (Ghiringhelli et al., 2015; Van Vechten, 1969) data sets. *Haberman* is a data set on medical case studies describing the survival of patients who had undergone surgery for breast cancer between 1958 and 1970 (Haberman, 1976). X consists of the age of the patient at time of operation, the patient's year of operation and the number of positive axillary nodes detected. Y is the survival status, which is binary and divided into longer or at most five years ($X \rightarrow Y$). The *Iris* data set contains data about three types of the Iris plant (Y) and four features dependent on which the type can be determined (Fisher, 1936). Next, we extract four cause-effect data sets from the *Mammals* data set (Heikinheimo et al., 2007), which consists of both climate data and presence records of 121 mammal species over 2 183 areas of 50×50km in Europe. We assume that elevation, precipitation, average temperature and the annual temperature range (X) cause the presence of a mammal and not contrarily. We created three data sets, *Canis*, *Lepus* and *Martes*, each containing locations of different types of the named species and one data set containing all three of them. Last, we created a data set based on the octet data set (Ghiringhelli et al., 2015; Van Vechten, 1969). Marx and Vreeken (2017) created 10 univariate cause effect pairs based on the data set that had all the same effect, which we combined to a single multivariate data set.

We compare Crack$_N$ with LTR (Janzing et al., 2010), Ergo (Vreeken, 2015) and Origo (Budhathoki and Vreeken, 2017b). Ergo and LTR do not consider categoric data, and are hence not applicable on all data sets. In addition, LTR is only applicable to strictly multivariate data sets. Crack$_N$ is applicable to all data sets, and infers 15/17 causal directions correctly, by which it has an overall accuracy of 88.2%. Importantly, the two wrong decisions have low confidences compared to the correct inferences.

In addition, we conduct an experiment to check whether or not our result is influenced by redundant variables within X or Y. Hence, we first apply a standard redundancy test (R, Hmisc, redun) to omit redundant attributes within X or Y ($R^2 \geq 0.95$). After the reduction step, we apply Crack$_N$ to the non-redundant pairs. As result, we found that the *Climate* cause-effect pair indeed contained redundant information and was inferred correctly after removing the redundant variables. For all other pairs, the prediction did not change. Hence, applying Crack$_N$ after redundancy correction leads to an accuracy of 94.4%.

8.5 Conclusion

We considered the problem of inferring the causal direction from the joint distribution of two univariate or multivariate random variables X and Y, consisting of single-, or mixed-type data. We pointed out weaknesses of known causal indicators and proposed the normalized causal indicator for mixed-type data and data with highly unbalanced domains. Further, we suggested a practical two-part MDL encoding based on classification and regression trees to instantiate the absolute and normalized causal indicators and provide Crack, a fast greedy heuristic to efficiently approximate the optimal MDL score.

In the experiments, we evaluated the advantages of our proposed causal indicators and gave advice on when to use them. On real-world benchmark data, we are on par with the state-of-the-art for univariate continuous data and beat the state-of-the-art on multivariate data with a wide margin. For future work, we aim to investigate the application of Crack for the discovery of causal networks as well as its application to biological networks.

Chapter 9

Conclusion

In this thesis, we focused on three different aspects in the broader research area of causal discovery. These concern the faithfulness assumption, conditional independence testing, and cause-effect inference. We formulated a research question for each of these topics, which defined the general problems that we aimed to solve. In the following, we will discuss the progress that we made towards answering these research questions, point out aspects that we did not solve yet, and examine possible avenues for future work.

The first research question that we formulated was concerned with the faithfulness assumption. We criticized that this essential assumption can be violated by different generating mechanisms. Our goal was to infer causal structures even if faithfulness is violated, and thus, classical approaches fail to infer the true causal graph. We formalized this research question as follows.

Question 1 *How can we discover causal DAGs in the presence of faithfulness violations induced by xor-type relations?*

To tackle this problem, we first thoroughly analyzed the conditional independence statements induced by xor relations. After that, we devised a weaker faithfulness assumption, 2-adjacency faithfulness, which is, opposed to normal faithfulness, not violated by such mechanisms. We further pointed out the difficulties such mechanisms impose for inferring the DAG structure and provided a sound inference rule to approach this problem. This new inference rule allows us to infer the causal DAG within such an xor relation if it appears in a larger graph. As a proof of concept, we additionally provide a sound adaptation of the GS algorithm to discover the Markov blanket of a target node under strictly weaker assumptions than faithfulness.

Although these contributions are important steps towards answering Question 1, the last step, the development of a sound causal discovery algorithm is yet to be done. A possible instantiation of such an algorithm could be achieved by extending existing constraint-based algorithms such as the PC algorithm (Spirtes et al., 2000) or score-based algorithms such as GES (Chickering, 2002) to our setting. For both, the main challenge would be to efficiently discover strict 2-associations and develop a smart procedure to infer the edge directions. The second part is especially difficult, considering we might not be able to infer the skeleton structure for all strict 2-associations. Another open problem is to extend our framework to the setup in which causal sufficiency does not hold, i.e., there might be unobserved confounders, or relax the assumption that all causal relations are acyclic. Besides, an interesting future project would be to evaluate how frequently such faithfulness violations occur in real data and evaluate our algorithm on such data.

In the second part of this thesis, we considered the problem of conditional independence testing, which lies at the core of causal discovery and largely influences the accuracy of causal discovery algorithms. Within the broad topic of independence testing, we considered the following research question.

Question 2 *How can we detect (conditional) dependencies among mixed-type random variables that can be discrete, continuous or a mixture of both?*

Towards this goal, we first developed a conditional independence criterium, SCCI, for discrete data based on stochastic complexity, which approximates algorithmic conditional mutual information. In addition, we proved that SCCI is a consistent estimator of CMI defined through Shannon entropy and empirically evaluated the sample complexity of SCCI on this task. Further, we benchmarked SCCI against state-of-the-art conditional independence tests in a causal discovery setup. We noticed that compared to its competitors, SCCI is able to pick up more true dependencies while not compromising the false discovery rate. As a follow-up project, we proposed a more general CMI estimator that can be applied to discrete, continuous and discrete-continuous mixture random variables. We proved that we can consistently estimate CMI for such data using adaptive histogram models. Through evaluations on synthetic data, we showed that among the tested estimators, our estimator is the only one that converges to the true CMI estimate on mixture data.

Despite these advancements, we did not entirely reach our goal yet. Being able to consistently estimate CMI is an important step towards independence testing. However, as we evaluated in Chapter 3, the plug-in estimator for CMI on discrete data tends to overestimate dependencies and hence, is almost never completely zero for independent variables. Thus, to reliably distinguish dependencies from independencies, we need to correct for this bias. We evaluated multiple possible correction criteria towards this goal, but the ideal remains to be found. In addition, we might improve our results for independence testing by tailoring the histogram estimation more towards this objective. On a different note, also the efficiency of the algorithm still leaves room for improvement. It would be interesting to evaluate faster and more efficient search heuristics to see whether we can achieve equally good results with less computational effort.

In the last part of this thesis, we considered the problem of telling cause from effect given an i.i.d. sample of numeric, continuous or mixed-type data. Being able to infer the causal direction in such a scenario helps to distinguish between Markov equivalent DAGs, and thus, enables us to infer the correct DAG structure. We formulated this objective in our third research question.

Question 3 *How can we distinguish between the two Markov equivalent DAGs $X \to Y$ and $Y \to X$, and do so with guarantees?*

To approach this problem, we build upon the algorithmic independence of conditionals, which states that the factorization of the joint distribution w.r.t. the true DAG is the shortest one (Janzing and Schölkopf, 2010). We first established a direct link between this postulate, which is formulated in terms of Kolmogorov complexity, and two-part MDL approximations of it. As a practical instantiation of this framework, we modelled dependencies between cause and effect for univariate numeric i.i.d. data using local and global regression functions. To efficiently estimate these functions from an empirical sample, we proposed Slope. Although performing well in practice, Slope does, however, not come with guarantees besides approximating the AIC postulate. Hence, we followed-up on Slope with Sloppy, for which we derived identifiability results for L_0-regularized regression functions and even managed to outperform Slope in multiple settings. Subsequently, we extended the ideas presented in Chapter 6 to multivariate mixed-type data and proposed one of the first methods for cause-effect inference that can be applied to such a general setting. Throughout the empirical evaluations for all these methods, we were able to show that our proposed algorithms are among the state-of-the-art on both synthetic data, as well as on real-world benchmark data.

Despite the work presented in this thesis, the key aspects of Slope and Sloppy have been further developed to infer the complete causal DAG among a set of random variables $X_1, \ldots, X_m$ (Mian et al., 2021). The corresponding algorithm Globe is able to infer the full DAG structure and not only the Markov equivalence class, as opposed to constraint-based methods. Another avenue for future improvements would be to relax the assumption of causal sufficiency or consider different generating mechanisms like the location-scale setup, in which Sloppy was not able to pick up the correct causal direction.

The research questions we tried to answer in this thesis have the overarching goal of bringing causal discovery closer to the application to real-world data. Naturally, part of this process is to develop algorithms that have only lightweight assumptions, like the one presented in Part I, are robust and applicable to different data types. MDL-based approaches, as the ones presented in Parts II and III, intrinsically aim towards robust methods by considering both the complexity of a model and how well it can fit the data. Simply put, these methods were developed in the spirit of Occam's Razor: if two models perform equally well, we should prefer the simpler one. Towards applicability on diverse data sets, we proposed two approaches (in Chapters 4 and 8), which are not limited to a specific data type, but can be applied in a mixed setting. To conclude, we hope that this thesis provided a small stepping stone towards more robust causal discovery methods that can be applied in a real-world setting.

Appendices

A.1 Example 2.2 in Detail

As described in Section 2.5, we can generate a DAG of the form $X \rightarrow Y \leftarrow Z$ and $W \rightarrow Y$ s.t. X, Y and Z form a minimal unfaithful triple and $W \not\perp\!\!\!\perp_P Y$ as follows. We generate X, Z, W and E independently, with X and Z as fair coins, W as a coin with $P(W = 1) = p$, where $0 < p < 1$ and E (the noise variable) as a biased coin with $P(E = 1) = q$, $0 < q < \frac{1}{2}$. With $q > 0$, we ensure that the function is non-deterministic. Further, we generate Y as

$$Y := ((X \oplus Z) \wedge W) \oplus E\ .$$

We will obtain that $P(Y = 1) = q + \frac{p}{2} - pq$. Further, we can calculate that $P(X = 1, Y = 1) = \frac{1}{2} P(Y = 1) = P(X = 1) \cdot P(Y = 1)$. Also, $P(X = 1, Y = 0) = P(X = 1) \cdot P(Y = 0)$, which means that they are marginally independent. The same holds for Z and Y. If we calculate the probability for all three variables, we get that $P(X = 0, Z = 1, Y = 1) = \frac{p+q-2pq}{4}$ and $P(X = 0, Z = 1) \cdot P(Y = 1) = \frac{1}{4} P(Y = 1)$. Hence, we need to solve

$$\begin{aligned} P(X = 0, Z = 1, Y = 1) &= P(X = 0, Z = 1) \cdot P(Y = 1) \\ \Leftrightarrow p + q - 2pq &= q + \frac{p}{2} - pq \\ \Leftrightarrow p - pq &= \frac{p}{2}\ . \end{aligned}$$

The only solutions are $p = 0$ or $q = \frac{1}{2}$, which we excluded. Hence, $Y \not\perp\!\!\!\perp_P \{X, Z\}$ and by weak union also $Y \not\perp\!\!\!\perp_P X \mid Z$, as well as $Y \not\perp\!\!\!\perp_P Z \mid X$. Since we know by assumption that $X \perp\!\!\!\perp_P Z$ we can conclude from Lemma 2.1 that also $X \not\perp\!\!\!\perp_P Z \mid Y$, which means that $\{X, Y, Z\}$ from a minimal unfaithful triple since W will also not cancel out any of these conditional dependencies. Next, we also find that $W \not\perp\!\!\!\perp_P Y$, since $P(W = 1, Y = 1) = \frac{p}{2}$, which is only equal to $P(W = 1) \cdot P(Y = 1)$, if $p = 0$, $p = 1$ or $q = \frac{1}{2}$, which we excluded, and hence $W \not\perp\!\!\!\perp_P Y$. Last, we need to show that $X \not\perp\!\!\!\perp_P W \mid \{Y, Z\}$ and that $Z \not\perp\!\!\!\perp_P W \mid \{X, Y\}$. We can write

$$P(X, W \mid Y, Z) = \frac{P(X, W, Y, Z)}{P(Y, Z)} \,.$$

To show conditional dependence, this value has to be different from $P(X \mid Y, Z) \cdot P(W \mid Y, Z)$. Consider the case where all variables are equal to one. Hence, we get that

$$\begin{aligned} P(X = 1, W = 1, Y = 1, Z = 1) &= \frac{pq}{4} \,, \\ P(X = 1, Y = 1, Z = 1) &= \frac{q}{4} \,, \\ P(W = 1, Y = 1, Z = 1) &= \frac{p}{4} \,. \end{aligned}$$

Since we know that $P(Y = 1, Z = 1) = P(Y = 1)/2$, we thus need to solve

$$pq = \frac{pq}{2P(Y = 1)} \,.$$

This equation can only be true if p or $q = 0$, i.e., the system is either independent of W or deterministic, $p = 1$ or $q = \frac{1}{2}$, which we all excluded by assumption. Hence, $X \not\perp\!\!\!\perp_P W \mid \{Y, Z\}$. The dependence between Z and W given X and Y can be derived in the same way.

A.2 2-Orientation Faithfulness and Sparsest Markov Representation

In this section, we briefly discuss the connection of our new assumptions to approaches based on the sparsest Markov representation (SMR) (Raskutti and Uhler, 2018) which is also referred to as frugality (Forster et al., 2017), which we discussed in the related work section. A graph G^* satisfies the SMR assumption if every graph G that fulfils the Markov property and is not in the Markov equivalence class of G^* contains more edges than G^*. Here we will not discuss the SMR assumption in further detail, but focus on the suggested causal discovery algorithm under the SMR assumption, which is called the Sparsest Permutation (SP) algorithm.

To explain the SP algorithm, we need to define a DAG G_π, w.r.t. a permutation π. A DAG G_π consists of vertices $\boldsymbol{V}$ and directed edges E_π, where an edge from the j-th node $\pi(j)$ according to permutation π to node $\pi(k)$ is in E_π if and only if $j < k$ and

$$X_{\pi(j)} \not\perp\!\!\!\perp_P X_{\pi(k)} \mid \{X_{\pi(1)}, X_{\pi(2)}, \ldots, X_{\pi(k-1)}\} \backslash \{X_{\pi(j)}\} ,$$

where $X_{\pi(j)}$ refers to the j-th random variable according to permutation π. Based on this definition, the SP algorithm constructs a graph G_π for each possible permutation and selects that permutation π^* for which G_{π^*} contains the fewest edges. This permutation π^* is also called minimal or a minimal permutation, if it is not unique.

Although this procedure might be very slow in practice, it is theoretically appealing. In particular, we conjecture that it can identify the collider pattern even if strict 2-associations are included, if 2-orientation faithfulness holds. Here, we will not provide a proof for this conjecture, but give some evidence by discussing the behaviour of the SP algorithm on an example graph.

Consider the graph provided in Figure 2.5(a) again. For this example, we assume that $\boldsymbol{V}$ does not consist of any further vertices than the four shown in the graph. We will show that all permutations π that are minimal have in common that $\pi(4) = Y$. W.l.o.g. let $\pi(1) = X, \pi(2) = Z$ and $\pi(3) = W$, then G_π only contains the three correct edges, which are:

$$\begin{aligned} \pi(1) \to \pi(4) &: X \not\perp\!\!\!\perp_P Y \mid \{Z, W\} \\ \pi(2) \to \pi(4) &: Z \not\perp\!\!\!\perp_P Y \mid \{X, W\} \\ \pi(3) \to \pi(4) &: W \not\perp\!\!\!\perp_P Y \mid \{X, Z\} \end{aligned}$$

and we do not add any superfluous edges, as

$$\begin{aligned} \pi(1) \to \pi(2) &: X \perp\!\!\!\perp_P Z \mid \emptyset \\ \pi(1) \to \pi(3) &: X \perp\!\!\!\perp_P W \mid Z \\ \pi(2) \to \pi(3) &: Z \perp\!\!\!\perp_P W \mid X \ . \end{aligned}$$

If we would pick a permutation π' in which we flip for example W and Y such that Y is no longer the node assigned to the highest number in the permutation, i.e., $\pi'(3) = Y$ and $\pi'(4) = W$, we will find more edges and thus not a minimal graph anymore. In particular, we get that

$$\begin{aligned}
\pi'(1) \to \pi'(3) &: X \not\perp\!\!\!\perp_P Y \mid \{Z\} \\
\pi'(2) \to \pi'(3) &: Z \not\perp\!\!\!\perp_P Y \mid \{X\} \\
\pi'(3) \to \pi'(4) &: Y \not\perp\!\!\!\perp_P W \mid \{X, Z\} \\
\pi'(1) \to \pi'(4) &: X \not\perp\!\!\!\perp_P W \mid \{Z, Y\} \\
\pi'(2) \to \pi'(4) &: Z \not\perp\!\!\!\perp_P W \mid \{X, Y\}
\end{aligned}$$

and thus the graph according to this permutation contains two edges more than for permutation π. The main point is that we are now allowed to condition on Y, which opens the paths between X or Z and W. Similarly, assume that we put X as the last node and get the order $\pi'(1) = Z, \pi'(2) = W, \pi'(3) = Y$ and $\pi'(4) = X$, for which

$$\begin{aligned}
\pi'(1) \to \pi'(2) &: Z \perp\!\!\!\perp_P W \mid \emptyset \\
\pi'(1) \to \pi'(3) &: Z \perp\!\!\!\perp_P Y \mid \{W\} \\
\pi'(1) \to \pi'(4) &: Z \not\perp\!\!\!\perp_P X \mid \{W, Y\} \\
\pi'(2) \to \pi'(3) &: W \not\perp\!\!\!\perp_P Y \mid \{Z\} \\
\pi'(2) \to \pi'(4) &: W \not\perp\!\!\!\perp_P X \mid \{Z, Y\} \\
\pi'(3) \to \pi'(4) &: Y \not\perp\!\!\!\perp_P X \mid \{Z, W\}
\end{aligned}$$

and hence, we again find four edges, which is one more than for π. Also, if $\pi'(1) = Y$, we can use it in the conditional to find a dependence between X and Z, and at least one dependence between X or Z and W. Hence, at least for this example graph, the SP algorithm would infer a correct ordering.

An interesting avenue for future work would be to analyze whether it is possible to always detect the collider pattern also in larger graphs and triples that may or may not be shielded.

A.3 Proofs for Chapter 2

Theorem 2.2 *Assuming that the CMC holds, the orientation rule in Definition 2.11 is sound.*

Proof: *First, we derive a general statement about the relations between* $\mathbf{X}$ *and* $\mathbf{Z}$ *without further specifying the role of* Y. *In particular, we show that* *there always exists a pair* $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ *s.t. w.l.o.g.*

$$X \perp\!\!\!\perp_G Z \mid Pa(X) \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\}) \;, \tag{1}$$

where $Pa(X) \subseteq \boldsymbol{V} \backslash \boldsymbol{Z}$*. Due to acyclicity, there has to exist a node in* $\boldsymbol{X} \cup \boldsymbol{Z}$*, say* X*, that is not an ancestor of any node in* $(\boldsymbol{X} \cup \boldsymbol{Z}) \backslash \{X\}$ *and hence* $(\boldsymbol{X} \cup \boldsymbol{Z}) \backslash \{X\} \subseteq Nd(X)$*. By the local Markov condition, we get that* $X \perp\!\!\!\perp_G (\boldsymbol{X} \cup \boldsymbol{Z}) \backslash \{X\} \mid Pa(X)$*. Thus, by weak union,*

$$X \perp\!\!\!\perp_G Z \mid Pa(X) \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\}) \;,$$

for any $Z \in \boldsymbol{Z}$*. Further,* $\boldsymbol{Z} \cap Pa(X) = \emptyset$*, as by assumption no pair of nodes* $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ *is adjacent in* G*.*

Since $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ *and* $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$*, we know that* Y *is at least adjacent to one node in* $\boldsymbol{X}$ *and one node in* $\boldsymbol{Z}$*. Hence,* Y *can take the following roles:*

a) Y *is a descendent of each node in* $\boldsymbol{X} \cup \boldsymbol{Z}$*, that is,* $\boldsymbol{X} \to Y \leftarrow \boldsymbol{Z}$*,*
b) Y *is a non-descendent of each node in* $\boldsymbol{X} \cup \boldsymbol{Z}$ *and*
c) Y *is a descendent of at least one node in* $\boldsymbol{X} \cup \boldsymbol{Z}$ *and a non-descendent of at least one node in* $\boldsymbol{X} \cup \boldsymbol{Z}$*.*

The first statement corresponds to the graph structure implied by rule i) and any possible structure from the latter two is implied by the probabilities found in rule ii). To show these two implications hold, we do a proof by contraposition for each rule.

Hence, to show rule i), we need to prove that if the graph structure is not a collider—i.e., Y *takes one of the roles described in b) or c)—then there exists a pair* $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ *and there exists a subset* $\boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, Z\}$ *s.t.*

$$X \perp\!\!\!\perp_P Z \mid \boldsymbol{S} \cup \{Y\} \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\}) \;.$$

First, consider all graphs in which Y *is a non-descendent of each node in* $\boldsymbol{X} \cup \boldsymbol{Z}$ *as described in b). We know from statement (*1*) that, w.l.o.g., there exists a pair* $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ *for which* $X \perp\!\!\!\perp_G Z \mid Pa(X) \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$*. Since* $Y \in Nd(X)$*, we will also find that* $X \perp\!\!\!\perp_G Z \mid Pa(X) \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\}) \cup \{Y\}$*, where* $Pa(X)$ *does not include* X *or* Z*. Thus, by CMC we found the required independence. For the cases described in c), again assume that* X *is not an ancestor of any node in* $(\boldsymbol{X} \cup \boldsymbol{Z}) \backslash \{X\}$*. To conclude the same statement as previously, we show that* X *has to be in* $De(Y)$ *and thus* $Y \in Nd(X)$*. We do this by deriving a contradiction: assume* $X \in Nd(Y)$*. If* $\boldsymbol{X}$ *consists only of the single node* X*, then* X *has to be adjacent to* Y*,* $X \in Pa(Y)$ *and hence* $X \to Y$ *in* G*. Thus,* Y *(and hence* X*) has to be an ancestor of at least one node in* $\boldsymbol{Z}$*, by assumption (*Y *is a non-descendent of at least one node in* $\boldsymbol{X} \cup \boldsymbol{Z}$*), which is a contradiction. Similarly, if* $\boldsymbol{X}$ *contains a second node,* X'*, we know by assumption that* $X' \in Nd(X)$*. We also know that the triple* $\{X, X', Y\}$ *has to* *contain a collider. X cannot be the collider, since $X \notin De(Y)$ and also X' cannot be the collider since $X \notin An(X')$. Hence, Y has to be the collider on the path $\langle X, Y, X' \rangle$. As above, at least one node $Z \in \boldsymbol{Z}$ has to be a descendent of Y, by assumption and thus, $X \in An(Z)$, which is a contradiction.*

Last, we prove that the implication in rule ii) holds. Thus, by contraposition, we need to show that if $\boldsymbol{X} \rightarrow Y \leftarrow \boldsymbol{Z}$, then there exists a pair $X, Z \in \boldsymbol{X} \times \boldsymbol{Z}$ s.t. X is conditionally independent of Z given a subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $(\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$ but does not contain Y. From statement (1) there exists a pair $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ that is d-separated given $Pa(X) \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$. Since Y cannot be in $Pa(X)$ due to acyclicity, we showed that there exists such a pair of nodes X, Z that can be rendered conditionally independent by a subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $(\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$ but does not contain Y (after applying CMC), which concludes the proof. □

Corollary 2.1 *Given $M := (G, \boldsymbol{V}, P)$ with $Y \in \boldsymbol{V}$ and $\boldsymbol{X}, \boldsymbol{Z} \subseteq \boldsymbol{V}$, where $\boldsymbol{X} \cap \boldsymbol{Z} = \emptyset$, $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$, $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$ and no pair of nodes $(X, Z) \in \boldsymbol{X} \times \boldsymbol{Z}$ is adjacent. Assuming that CMC holds, we can detect if condition i) or ii) of 2-orientation faithfulness fails on the triple $\{\boldsymbol{X}, Y, \boldsymbol{Z}\}$.*

PROOF: *Since we know that $Y \overset{s}{-}_{\leq 2} \boldsymbol{X}$ and $Y \overset{s}{-}_{\leq 2} \boldsymbol{Z}$, we can conclude that, as in the proof of Theorem 2.2, Y can take three different roles w.r.t. $\boldsymbol{X}$ and $\boldsymbol{Z}$, where role a) corresponds to condition i) in 2-orientation faithfulness and rule i) in the orientation rule and roles b) and c) correspond to condition ii) and rule ii).*

Now assume that condition i) in 2-orientation faithfulness fails, that is, the true graph can be described by role a), but there exists a pair $X \in \boldsymbol{X}$ and $Z \in \boldsymbol{Z}$, for which X is independent of Z given a subset of $\boldsymbol{V} \backslash \{X, Z\}$ that contains $Y \cup (\boldsymbol{X} \backslash \{X\}) \cup (\boldsymbol{Z} \backslash \{Z\})$. If this is the case, we cannot apply rule i) of our orientation rule. In addition, we showed in Theorem 2.2 that for a graph as described by a) rule ii) can never apply. Thus, we can detect this failure of condition i) in 2-orientation faithfulness by noticing that neither rule i) nor ii) of our orientation rule applies.

Next, assume condition ii) in 2-orientation fails. This means that we cannot apply rule ii) of the orientation rule. Again, we showed that for such graphs Y takes either role b) or c), in which case orientation rule i) can never apply. Hence, we can detect if condition ii) in 2-orientation faithfulness fails, since none of the two conditions in our orientation rule are met. □

Theorem 2.3 *Let $M = (G, \boldsymbol{V}, P)$, and assume 2-adjacency faithfulness, Assumption 2.1 and CMC hold. Algorithm 2.1 identifies $MB(T)$ for $T \in \boldsymbol{V}$.*

PROOF: *We follow the original correctness proof under the faithfulness assumption (Margaritis and Thrun, 2000), that consists of two main steps. First, we need to show that MB(T) $\subseteq \boldsymbol{S}$ after the grow phase and second, we need to ensure that all nodes in MB(T) stay in $\boldsymbol{S}$ during the shrink phase, while nodes not in MB(T) will be removed from $\boldsymbol{S}$ in the shrink phase.*

Grow phase: By assumption (2-adjacency faithfulness), for each node $X \in$ PC(T), T is either 1-associated to X, or there exists a set $\boldsymbol{X}$ that includes X such that $T \overset{s}{-}_2 \boldsymbol{X}$. If T is 1-associated to a node X, then $T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S}$, if $X \notin \boldsymbol{S}$, hence we will add those nodes. If T is strictly 2-associated to a set $\{X, Z\}$ then $T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S} \cup \{Z\}$ for all $\boldsymbol{S} \subseteq \boldsymbol{V} \backslash \{X, T, Z\}$. Thus, we also add X to $\boldsymbol{S}$, if $X \notin \boldsymbol{S}$ and afterwards also find that $T \not\perp\!\!\!\perp_P Z \mid \boldsymbol{S}$, if $Z \notin \boldsymbol{S}$, since $X \in \boldsymbol{S}$. Hence, all nodes in PC(T) will be added during the grow phase. Next, we need to consider the spouses of T that do not overlap with PC(T), hence might not have been added yet.[1] Since we know that eventually $\boldsymbol{S}$ will contain all children of T, we will afterwards also add the corresponding spouses. In particular, we need to consider two classes of spouses S: 1) Spouses that through a child node C are strictly 2-associated to T ($T \overset{s}{-}_2 \{C, S\}$). Those will be added due to the strict 2-association as explained above. 2) Spouses that are not involved in such a strict 2-association. For the latter, we find a conditional dependence between T and S by conditioning on the corresponding child node C (by Assumption 2.1), which will be in $\boldsymbol{S}$. A special case occurs if a child node C is strictly 2-associated to two spouses S_1 and S_2. Due to Assumption 2.1, T is dependent on S_1 if we condition on C and S_2, vice versa T is dependent on S_2 if we condition on C and S_1. Similarly to how we add strict 2-associations above, we will also first add one of the two and then the second one. Thus, after the grow phase, $\boldsymbol{S}$ will contain all elements of MB(T).

Shrink phase: Since it is possible that after the grow phase $\boldsymbol{S}$ is a superset of MB(T), we need to ensure that in the shrink phase all $W \notin$ MB(T) will be deleted from $\boldsymbol{S}$ and all $X \in$ MB(T) will stay in $\boldsymbol{S}$.

First, we show that no node $X \in$ MB(T) will be removed from $\boldsymbol{S}$. Assume X is the first element in MB(T) that we attempt to remove from $\boldsymbol{S}$. If $X \in$ PC(T), by definition of 2-adjacency faithfulness T is either 1-associated to X and hence, X will not be removed, or T is strictly 2-associated to a set $\boldsymbol{X} \subseteq$ MB(T) that contains X. W.l.o.g. let $\boldsymbol{X} = \{X, Z\}$, then $T \not\perp\!\!\!\perp_P X \mid \boldsymbol{S} \backslash \{X\}$, since $\boldsymbol{S}$ contains Z, and hence, X will not be removed from $\boldsymbol{S}$. If X is a spouse of T, there again exist two cases. Either T is strictly 2-associated to a set that contains X, in which case, X will not be removed from $\boldsymbol{S}$ as explained above, or T is not strictly 2-associated to a set that contains X. In the latter case, by Assumption 2.1, X is dependent on T conditioned on a subset of MB(T)$\backslash \{X\}$ and thus $X \not\perp\!\!\!\perp_P T \mid \boldsymbol{S} \backslash \{X\}$. In particular, this subset consists of the common

[1] There could be nodes that are spouses of T and in PC(T) at the same time, e.g., if T has two children X and Z, where Z is also a parent of X.

child C and in the special case that C is strictly 2-associated to X and a second spouse S, it also contains that second spouse S. Either way, those conditioning sets are contained in $\boldsymbol{S}$. Hence, X will not be removed from $\boldsymbol{S}$. In the following iterations, $\boldsymbol{S}$ will still contain $MB(T)$ and hence, we will also not remove a true element of $MB(T)$.

Last, assume $W \notin MB(T)$, but $W \in \boldsymbol{S}$ after the grow phase. Further, we can write $\boldsymbol{S}\backslash\{W\}$ as $MB(T) \cup \boldsymbol{Q}$, where $\boldsymbol{Q}$ contains all elements from $\boldsymbol{S}\backslash\{W\}$ that are not in $MB(T)$. Then, $T \perp\!\!\!\perp_G \{W\} \cup \boldsymbol{Q} \mid MB(T)$ and thus by weak union, $T \perp\!\!\!\perp_G W \mid MB(T) \cup \boldsymbol{Q}$, which implies $T \perp\!\!\!\perp_P W \mid \boldsymbol{S}\backslash\{W\}$ (by CMC). Hence, we delete each node in $\boldsymbol{S}$ that is not in $MB(T)$ in the shrink phase. □

A.4 Proofs for Chapter 3

Lemma 3.1 *For $n \geq 1$, the regret $\mathcal{C}_k^n$ of the multinomial stochastic complexity of a random variable with a domain size of $k \geq 2$ is log-concave in n.*

Proof: *To improve the readability of this proof, we write $\mathcal{C}_L^n$ as shorthand for $\mathcal{C}_{\mathcal{M}_L}^n$ of a random variable with a domain size of L. Since n is an integer, each $\mathcal{C}_L^n > 0$ and $\mathcal{C}_L^0 = 1$, we can prove Lemma 3.1, by showing that the fraction $\mathcal{C}_L^n / \mathcal{C}_L^{n-1}$ is decreasing for $n \geq 1$, when n increases. We know from Mononen and Myllymäki (2008) that $\mathcal{C}_L^n$ can be written as the sum*

$$\mathcal{C}_L^n = \sum_{k=0}^{n} m(k,n) = \sum_{k=0}^{n} \frac{n^{\underline{k}}(L-1)^{\overline{k}}}{n^k k!} \ ,$$

where $x^{\underline{k}}$ represent falling factorials and $x^{\overline{k}}$ rising factorials. Further, they show that for fixed n we can write $m(k,n)$ as

$$m(k,n) = m(k-1,n)\frac{(n-k+1)(k+L-2)}{nk} \ , \tag{2}$$

where $m(0,n)$ is equal to 1. It is easy to see that from $n = 1$ to $n = 2$ the fraction $\mathcal{C}_L^n / \mathcal{C}_L^{n-1}$ decreases, as $\mathcal{C}_L^0 = 1$, $\mathcal{C}_L^1 = L$ and $\mathcal{C}_L^2 = L + L(L-1)/2$. In the following, we will show the general case. We rewrite the fraction as follows.

$$\begin{aligned}\frac{\mathcal{C}_L^n}{\mathcal{C}_L^{n-1}} &= \frac{\sum_{k=0}^{n} m(k,n)}{\sum_{k=0}^{n-1} m(k,n-1)} \\ &= \frac{\sum_{k=0}^{n-1} m(k,n)}{\sum_{k=0}^{n-1} m(k,n-1)} + \frac{m(n,n)}{\sum_{k=0}^{n-1} m(k,n-1)} \qquad (3)\end{aligned}$$

Next, we will show that both parts of the sum in Equation 3 *are decreasing when* n *increases. We start with the left part, which we rewrite to*

$$\begin{aligned}\frac{\sum_{k=0}^{n-1} m(k,n)}{\sum_{k=0}^{n-1} m(k,n-1)} &= \frac{\sum_{k=0}^{n-1} m(k,n-1) + \sum_{k=0}^{n-1} \left(m(k,n) - m(k,n-1)\right)}{\sum_{k=0}^{n-1} m(k,n-1)} \\ &= 1 + \frac{\sum_{k=0}^{n-1} \frac{(L-1)^{\overline{k}}}{k!} \left(\frac{n^{\underline{k}}}{n^k} - \frac{(n-1)^{\underline{k}}}{(n-1)^k}\right)}{\sum_{k=0}^{n-1} m(k,n-1)} \ . \end{aligned} \tag{4}$$

When n *increases, each term of the sum in the numerator in Equation* 4 *decreases, while each element of the sum in the denominator increases. Hence, the whole term is decreasing. In the next step, we show that the right term in Equation* 3 *also decreases when* n *increases. It holds that*

$$\frac{m(n,n)}{\sum_{k=0}^{n-1} m(k,n-1)} \geq \frac{m(n,n)}{m(n-1,n-1)} \ .$$

Using Equation 2 *we can reformulate the term as follows.*

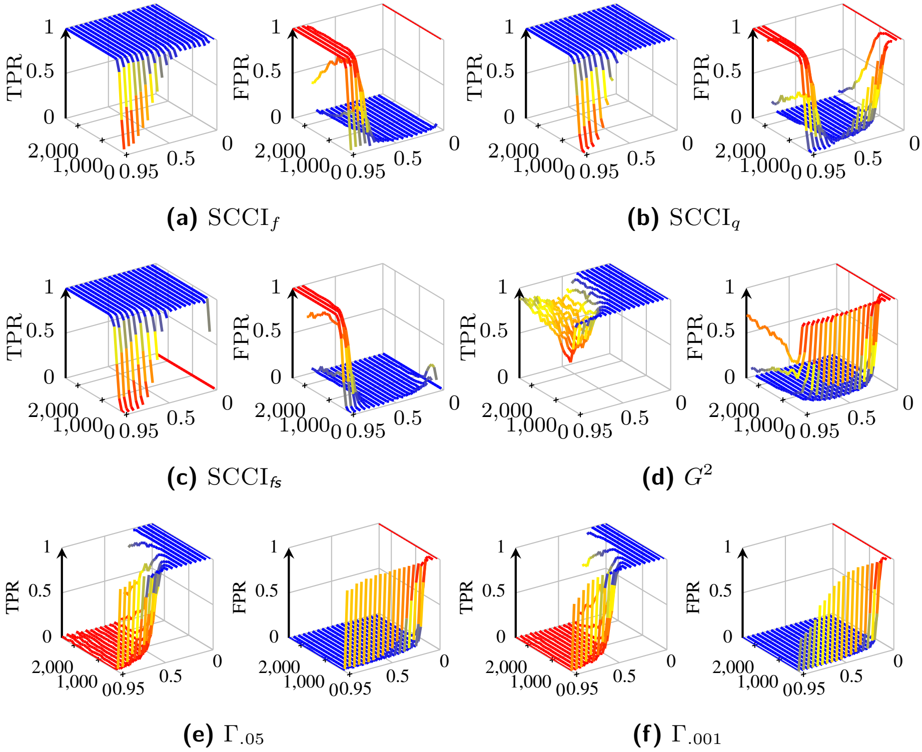
$$\frac{\frac{n+L-2}{n^2} m(n-1,n)}{m(n-1,n-1)} = \frac{n+L-2}{n^2}\left(1 + \frac{m(n-1,n) - m(n-1,n-1)}{m(n-1,n-1)}\right)$$

After rewriting, we have that $\frac{n+L-2}{n^2}$ *is definitely decreasing with increasing* n. *For the right part of the product, we can argue the same way as for Equation* 4. *Hence the whole term is decreasing, which concludes the proof.* □

A.5 Additional Experiments for Chapter 3

In this section, we provide more details to the true positive and false positive rates w.r.t. the experiments in Section 3.6.1, which we show in Figure 1. In addition, we also provide the results for SCCI_{fs} and CMI_{Γ} with $\alpha = 0.001$. Since we did not provide the accuracy of JIC for this experiment in the main body of Chapter 3, we plot the accuracy, true and false positive rates of JIC in Figure 3 and analyze those results at the end of this section.

From Figure 1, we see that SCCI_f and SCCI_{fs} perform best. Only for very high noise setups ($\geq 70\%$) they start to flag everything as independent. The G^2 test struggles with small sample sizes. It needs more than 500 samples and is inconsistent given more than 35% noise. Note that we forced G^2 to decide for every sample size, while the minimum number of samples recommended for G^2 on this data set would be 1 440, which corresponds to $10(|\mathcal{X}|-1)(|\mathcal{Y}|-1)|\mathcal{Z}|$ (Kalisch et al., 2012). Further, we observe that there is



(a) SCCI_f

(b) SCCI_q

(c) SCCI_{fs}

(d) G^2

(e) $\Gamma_{.05}$

(f) $\Gamma_{.001}$

Figure 1: True positive (TPR) and false positive rates (FPR) of SCCI_f, SCCI_q, SCCI_{fs}, G^2 and CMI_Γ with $\alpha = 0.05$ ($\Gamma_{.05}$) and $\alpha = 0.001$ ($\Gamma_{.001}$) for identifying d-separation. We use varying samples sizes (x-axis) and additive noise percentages (y-axis) as in Figure 3.5, where a noise level of 0.95 refers to 95% additive noise.

barely any difference between CMI_Γ using $\alpha = 0.05$ or $\alpha = 0.001$ as a significance level. After more than 20% noise has been added, CMI_Γ starts to flag everything as dependent.

Next, we also show the accuracy for identifying d-separation for CMI with zero as threshold in Figure 2. Overall, it performs very poorly, which raises from the fact that it barely finds any independence. In addition to the accuracy of CMI, we also plot the average value that CMI reports for the true positive case ($F \perp\!\!\!\perp T \mid \{D, E\}$), where it should be equal to zero. It can be seen that it is dependent on the noise level as well as the sample size. This could explain, why SCCI_f performs best on the d-separation data. Since the noise is uniform, the threshold for SCCI_f is likely to be higher the more noise has been added.

The JIC test has the opposite problem. For the d-separation scenario that we picked it is too restrictive and falsely detects independencies where the

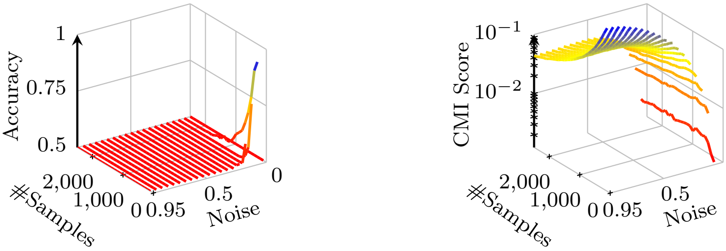


Figure 2: Accuracy of empirical CMI (left) and the average value of empirical CMI for the true independent case (right) for varying samples sizes and additive noise percentages. $\hat{I}(F;T \mid \{D,E\})$ is larger for small sample sizes and high noise settings.

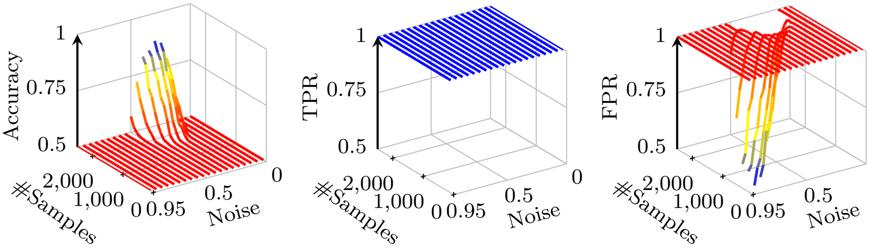


Figure 3: Accuracy, true positive (TPR) and false positive rates (FPR) of JIC for identifying d-separation. We use varying samples sizes and additive noise percentages, where a noise level of 0.95 refers to 95% additive noise.

ground truth is dependent, as shown in Figure 3. As the discrete version of JIC is calculated from the empirical entropies and a penalizing term based on the asymptotic formulation of stochastic complexity—i.e.,

$$\mathrm{JIC}(X;Y \mid Z) := \max\{\hat{I}(X;Y \mid Z) - \frac{(|\mathcal{X}|-1)(|\mathcal{Y}|-1)|\mathcal{Z}|}{2n}\log n, 0\}\,,$$

it penalizes quite strongly in our example since $|\mathcal{Z}| = 16$. As JIC is based on an asymptotic formulation of stochastic complexity, we expect it to perform better given more data.

A.6 Proofs for Chapter 4

Theorem 4.1 *Given a mixed random vector (X,Y,Z) with probability measure P_{XYZ}, $\lim_{v'\to 0}\lim_{n\to\infty} I^h(X;Y \mid Z) = I(X;Y \mid Z)$ almost surely, where n refers to the sample size and v' refers to the maximum of the histogram volumes* *for bins in B' (defined in Section 4.2.2).*

Proof: *To prove Theorem 4.1 we need several intermediate results. Lemma A.3 shows that a histogram results in a valid discretization as all terms corresponding to volumes in I^h cancel out, and hence I^h can be written as a sum of plug-in estimators of discrete entropies. Then, Lemma A.1 shows a classic result that the plug-in estimator of discrete entropies will converge to the true entropy almost surely. Further, we show in Lemma A.2 that as the volumes of histogram bins containing continuous values go to* 0*, the true entropies of the discretized variables (which are discretized by the histogram) converge to the true entropies of original variables.*

Lemma A.1 *Given a discrete random vector (X_d, Y_d, Z_d), the empirical entropy $\hat{H}$ converges to the true entropy as $n \to \infty$, i.e., $lim_{n\to\infty}\hat{H}(X_d, Y_d, Z_d) = H(X_d, Y_d, Z_d)$ almost surely (Antos and Kontoyiannis, 2001).*

Lemma A.2 *Given a random vector (X, Y, Z) that contains discrete-continuous mixture random variables, with bins $B = B' \cup B''$ and the resulting discretized random vector (X_d, Y_d, Z_d), where B'' contains discrete data points (of which every dimension has a discrete value) and $B' = B \setminus B''$, we have*

$$\lim_{v'\to 0} H(X_d, Y_d, Z_d) = H(X, Y, Z) \ ,$$

where $v' = \max_{B_j \in B'}(v(B_j))$.

Proof: *Firstly, it is well-known that this result holds if (X, Y, Z) is a continuous random vector (Cover and Thomas, 2012); then, if (X, Y, Z) contains mixture variables,*

$$H(X, Y, Z) = \lim_{v'\to 0} \sum_{B_j \in B'} \frac{P_{X_dY_dZ_d}}{v(B_j)} \log \frac{P_{X_dY_dZ_d}}{v(B_j)} + \sum_{B_j \in B''} \frac{P_{X_dY_dZ_d}}{v(B_j)} \log \frac{P_{X_dY_dZ_d}}{v(B_j)}$$

$$= \lim_{v'\to 0} H(X_d, Y_d, Z_d) \ ,$$

which concludes the proof. □

Definition A.1 *Given a random vector (X, Y, Z) that contains mixture variables, and an adaptive grid B, we define the discretized random variable X_d, Y_d, Z_d, with probability measure (probability mass function)*

$$P_{X_d,Y_d,Z_d}((j_1, j_2, j_3)) = \int_{B_j} \frac{d_{XYZ}}{dv} dv \ ,$$

where B_j denotes the jth bin of B.

Lemma A.3 *Given an m-dimensional random vector (X, Y, Z) that contains mixture variables with an unknown probability measure P_{XYZ}, a dataset $D = (x_i, y_i, z_i)_{i\in\{1,\ldots,n\}}$ generated by P_{XYZ}, a histogram model M, and corresponding discretized random vector (X_d, Y_d, Z_d), we have*

$$I^h(X, Y|Z) = \hat{H}(X_d, Z_d) + \hat{H}(Y_d, Z_d) - \hat{H}(X_d, Y_d, Z_d) - \hat{H}(Z_d) \ .$$

That is, the terms corresponding to volumes in I^h cancel out and our histogram model results a valid discretization.

Proof: *Denote the adaptive grid of histogram model M as B^{XYZ}, which is the Cartesian product of bins defined on X, Y, Z—i.e. $B^{XYZ} = B^X \times B^Y \times B^Z$, and denote the corresponding MLE of histogram density function as $f^h_{\hat{\theta}_{XYZ}}$. Further, define a function v_X, such that for each x_i in D, $v_X(x_i) = v(B^X_j)$ if $x_i \in B^X_j$, where B^X_j is a bin of B^X and v is defined based on the random variable X. Then, define $v_Y, v_Z, v_{XZ}, v_{YZ}, v_{XYZ}$ similarly. By the definition*

$$I^h(X, Y \mid Z) = H^h(X, Z) + H^h(Y, Z) - H^h(X, Y, Z) - H^h(Z) \ .$$

First consider $H^h(X, Z)$. We write $B^{XZ} = B^X \times B^Z$, with marginal density function $f^h_{\hat{\theta}_{XZ}}$. W.l.o.g. suppose that B_{XZ} consists of k bins, denoted as $B^{XZ}_j, j \in \{1, \ldots, k\}$. Then,

$$\begin{aligned}
H^h(X, Z) &= -\int_{\mathbb{R}^{m_X+m_Z}} f^h_{\hat{\theta}_{XZ}} \log f^h_{\hat{\theta}_{XZ}} dv \\
&= -\sum_{j=1}^{k} \int_{B^{XZ}_j} f^h_{\hat{\theta}_{XZ}} \log f^h_{\hat{\theta}_{XZ}} dv \\
&= -\sum_{j=1}^{k} c_j \log\left(\frac{c_j}{nv(B_j)}\right) \\
&= -\sum_{j=1}^{k} c_j \log\left(\frac{c_j}{n}\right) + \sum_{i=1}^{n} \log(v_{XZ}(x_i, z_i)) \\
&= \hat{H}(X_d, Z_d) + \sum_{i=1}^{n} \log(v_{XZ}(x_i, z_i)) \ ,
\end{aligned}$$

where c_j is the number of data points in B_j and $v_{XZ}(x_i, z_i) = v_X(x_i)v_Z(z_i)$. The remaining entropies can be calculated similarly. Hence, $I^h(X, Y \mid Z) =$

$\hat{H}(X_d, Z_d) + \hat{H}(Y_d, Z_d) - \hat{H}(X_d, Y_d, Z_d) - \hat{H}(Z_d)$, as the sum of the volume related terms is equal to zero. That is,

$$\sum_{i=1}^{n} \log(v_{XZ}(x_i, z_i)) + \sum_{i=1}^{n} \log(v_{YZ}(y_i, z_i))$$
$$- \sum_{i=1}^{n} \log(v_{XYZ}(x_i, y_i, z_i)) - \sum_{i=1}^{n} \log(v_Z(z_i)) = 0 \ .$$

□

To conclude the proof of Theorem 4.1, we link the above results:

$$\begin{aligned}
&\lim_{v' \to 0} \lim_{n \to \infty} I^h(X; Y \mid Z) \\
&= \lim_{v' \to 0} \lim_{n \to \infty} (H^h(X,Z) + H^h(Y,Z) - H^h(X,Y,Z) - H^h(Z)) \\
&= \lim_{v' \to 0} \lim_{n \to \infty} (\hat{H}(X_d, Z_d) + \hat{H}(Y_d, Z_d) - \hat{H}(X_d, Y_d, Z_d) - \hat{H}(Z_d)) \\
&= \lim_{v' \to 0} (H(X_d, Z_d) + H(Y_d, Z_d) - H(X_d, Y_d, Z_d) - H(Z_d)) \\
&= I(X; Y \mid Z) \ .
\end{aligned}$$

□

A.6.1 Additional Experiments for Chapter 4

In the following, we provide additional experiments to evaluate $I_{\mathcal{X}^2}$ and I_{SC}. To evaluate how well $I_{\mathcal{X}^2}$ and I_{SC} can identify conditional (in)dependencies, we evaluate both variants on various generating mechanisms that involve collider and non-collider structures. As in the causal discovery experiment, we set $\alpha = 0.01$ for $I_{\mathcal{X}^2}$, RCIT and RCoT.

Collider Structures We generate data according to a collider structure, which can be represented by a directed acyclic graph as, e.g. $X \to Z \leftarrow Y$. According to this structure, we model X and Y by some distribution and write Z as a non-deterministic function of X and Y. We generate data for different generating mechanisms, including two continuous and four mixed settings:

1. $X \perp\!\!\!\perp Y$ and X, Y are either drawn from $N(0,1)$ or from Unif$(-2, 2)$. Z is an additive function of polynomials up to degree three or the tangent function plus additive independent noise $N \sim N(0, 0.1)$—e.g. $Z = X^3 + \tan(Y) + N$. We pick the type of the distribution of X, Y, as well as the function type, uniform at random.

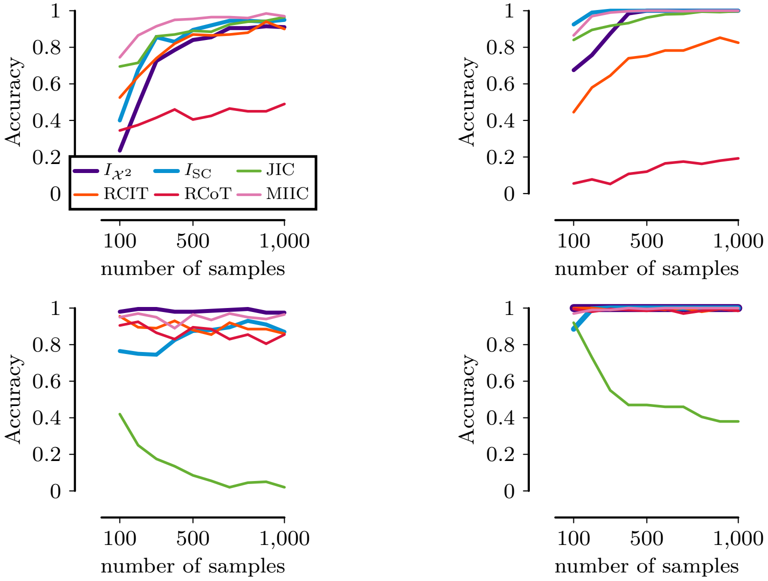


Figure 4: Accuracy for detecting continuous (left) and mixed-type (right) dependencies in collider structures (top) and independencies in non-collider structures (bottom) for different sample sizes.

2. X, Y are drawn from a standard Gaussian distribution, with $X \perp\!\!\!\perp Y$ and $Z = \text{sign}(X \cdot Y) \cdot \text{Exp}(1/\sqrt{2})$. Note that this is a generating mechanism that is likely to induce a faithfulness violation, as explained in Chapter 2.
3. $X, Y \sim N(0, 1)$ with $X \perp\!\!\!\perp Y$ and $Z = \text{sign}(X \cdot Y)$, where we randomly assign a $z \in \mathcal{Z}$ to 10% of the values in Z to make the function non-deterministic.
4. $X \sim N(0, 1)$, $Y \sim \text{Pois}(\lambda)$, with parameter λ selected uniformly at random from $\{1, 2, 3\}$. We generate Z as X modulo Y and assign 10% of the data points randomly.
5. X, Y are unbiased coins. $Z' = X \oplus Y$, where $\oplus$ denotes the xor operator. From Z' we calculate Z as $N(0, 0.1)$ if $Z' = 0$ and $\text{Pois}(5) \cdot N(0, 0.1)$ under the condition that $Z' = 1$.
6. We generate X, Y and Z' as above, but this time we generate Z as $\text{Pois}(5) + N(0, 0.1)$ if $Z' = 1$ and as $N(0, 0.1)$ if $Z' = 0$.

For each mechanism we generate 100 data sets and report the averaged results, separately for the continuous and mixed data, in Figure 4 (top). On the continuous data, both of our approaches perform on par with RCIT and JIC for more than 400 data points, whereas MIIC has a slightly better performance and RCoT is not able to detect the dependence for the sign function and hence has an accuracy of about 50%. Since the functions for mixed data include an xor and the modulo operator, it is difficult to treat all discrete variables as ordinal and hence RCIT only reaches up to 80% accuracy—which is mostly due to an xor determining the scaling of a Gaussian distributed variable. On the other hand, both of our tests perform very well and only need 400 samples to obtain an accuracy close to 100%. JIC and MIIC perform on par with our tests.

Non-Collider Structures Similar to collider structures, there also exist non-collider structures of the form $X \rightarrow Z \rightarrow Y$ or $X \leftarrow Z \rightarrow Y$. In both cases, the ground truth is that $X \perp\!\!\!\perp Y \mid Z$. To simulate data according to these graphs, we consider two continuous mechanisms based on polynomial functions and two mixed generating mechanisms:

1. $X \sim N(0,1)$, Z is an additive noise function of X and Y is an additive noise function of Z. The functions can be polynomials up to degree three or the tangent function.
2. $Z \sim N(0,1)$, X and Y are independent additive noise functions of Z, as defined above.
3. X, Y and Z are generated as in Experiment IV.
4. X and Y are generated according to Experiment II and $Z \sim N(\mu, x)$ for $X = x$ and $\mu \in [-4, 4]$.

In essence, Figure 4 (bottom) shows that $I_{\mathcal{X}^2}$ has an almost perfect accuracy for the continuous and mixed data, whereas RCIT and RCoT fail to detect up to 20% of the independencies for continuous data, MIIC does not detect up to 11% and JIC seems to generally overestimate dependencies for those test cases. If we consider these results in comparison to the results for detecting dependencies for the collider setting, we suspect that both MIIC and JIC have a larger tendency to falsely detect dependencies, while our approach is more conservative and hence needs more samples to detect true dependencies. Despite an almost perfect performance on mixed data, I_{SC} is not as accurate on the purely continuous data. This is due to the way we compute the regret terms. In particular, for linear functions that are almost deterministic, which describes about 1/4 of all dependencies, the regret tends to be too lenient.

References

N. H. Abel. Démonstration de l'impossibilité de la résolution algébrique des équations générales qui passent le quatrieme degré. *Journal für die reine und angewandte Mathematik*, 1:65–96, 1826.

H. Akaike. Information measures and model selection. *Int Stat Inst*, 44:277–291, 1983.

H. Andersen. When to Expect Violations of Causal Faithfulness and Why it Matters. *Philosophy of Science*, 80(5):672–683, 2013.

D. Anderson, K. Burnham, and W. Thompson. Null Hypothesis Testing: Problems , Prevalence , and an Alternative. *The Journal of Wildlife Management*, 64(4):912–923, 2000.

A. Antos and I. Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19(3-4):163–193, 2001.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):289–300, 1995. ISSN 0035-9246.

W. Bergsma. Testing conditional independence for continuous random variables. *Report Eurandom*, 2004-048, 2004.

S. Bernstein. *Theory of Probability.* Moscow, 1927.

P. Blöbaum, D. Janzing, T. Washio, S. Shimizu, and B. Schölkopf. Cause-Effect Inference by Comparing Regression Errors. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

P. Bloem and S. de Rooij. Large-scale network motif analysis using compression. *Data Mining and Knowledge Discovery*, 34(5):1421–1453, 2020.

K. Budhathoki and J. Vreeken. MDL for Causal Inference on Discrete Data. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 751–756. IEEE, 2017a.

K. Budhathoki and J. Vreeken. Origo: causal inference by compression. *Knowledge and Information Systems*, pages 1–23, 2017b.

P. Bühlmann, J. Peters, J. Ernest, et al. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

V. Cabeli, L. Verny, N. Sella, G. Uguzzoni, M. Verny, and H. Isambert. Learning clinical networks from medical records based on information estimates in mixed-type data. *PLOS Computational Biology*, 16(5):e1007866, 2020.

G. J. Chaitin. A Theory of Program Size Formally Identical to Information Theory. *Journal of the ACM*, 22(3):329–340, 1975.

Z. Chen, K. Zhang, and L. Chan. Nonlinear Causal Discovery for High Dimensional Data: A Kernelized Trace Method. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1003–1008. IEEE, 2013.

D. M. Chickering. Optimal Structure Identification With Greedy Search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

G. A. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

A. P. Dawid. Conditional Independence in Statistical Theory. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(1):1–15, 1979.

D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

P. Forré and J. M. Mooij. Causal Calculus in the Presence of Cycles, Latent Confounders and Selection Bias. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI, 2019.

M. Forster, G. Raskutti, R. Stern, and N. Weinberger. The Frugal Inference of Causal Relations. *The British Journal for the Philosophy of Science*, 69 (3):821–848, 2017.

S. Frenzel and B. Pompe. Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, 99(20):204101, 2007.

W. Gao, S. Oh, and P. Viswanath. Breaking the Bandwidth Barrier: Geometrical Adaptive Entropy Estimation. In *Advances in Neural Information Processing Systems*, pages 2460–2468. Curran Associates, Inc., 2016.

W. Gao, S. Kannan, S. Oh, and P. Viswanath. Estimating mutual information for discrete-continuous mixtures. In *Advances in Neural Information Processing Systems*, pages 5986–5997, 2017.

D. Geiger, T. Verma, and J. Pearl. Identifying Independence in Bayesian Networks. *Networks*, 20(5):507–534, 1990.

L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler. Big data of materials science: Critical role of the descriptor. *Physical Review Letters*, 114(10):1–5, 2015.

B. Goebel, Z. Dawy, J. Hagenauer, and J. C. Mueller. An approximation to the distribution of finite sample size mutual information estimates. In *IEEE International Conference on Communications*, volume 2, pages 1102–1106. IEEE, 2005.

B. R. Goldsmith, M. Boley, J. Vreeken, M. Scheffler, and L. M. Ghiringhelli. Uncovering structure-property relationships of materials by subgroup discovery. *New Journal of Physics*, 19(1):013031, 2017.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, pages 585–592, 2008.

P. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

S. J. Haberman. Generalized residuals for log-linear models. In *Proceedings of the International Biometrics Conference*, pages 104–122, 1976.

Y. Han, J. Jiao, and T. Weissman. Adaptive estimation of Shannon entropy. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1372–1376. IEEE, 2015.

H. Heikinheimo, M. Fortelius, J. Eronen, and H. Mannila. Biogeography of European land mammals shows environmentally distinct and spatially coherent cluster. *Journal of Biogeography*, 34:1053–1064, 2007.

G. Hesslow. Two Notes on the Probabilistic Approach to Causality. *Philosophy of science*, 43(2):290–292, 1976.

P. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

S. Hu, Z. Chen, V. Partovi Nia, L. CHAN, and Y. Geng. Causal Inference and Mechanism Clustering of A Mixture of Additive Noise Models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5212–5222. Curran Associates, Inc., 2018.

T. Inazumi, T. Washio, S. Shimizu, J. Suzuki, A. Yamamoto, and Y. Kawahara. Discovering causal structures in binary exclusive-or skew acyclic models. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 373–382. AUAI, 2011.

D. Janzing and B. Schölkopf. Causal Inference Using the Algorithmic Markov Condition. *IEEE Transactions on Information Technology*, 56(10):5168–5194, 2010.

D. Janzing and B. Steudel. Justifying Additive Noise Model-Based Causal Discovery via Algorithmic Information Theory. *Open Systems and Information Dynamics*, 17(2):189–212, 2010.

D. Janzing, P. Hoyer, and B. Schölkopf. Telling cause from effect based on high-dimensional observations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 479–486. JMLR, 2010.

D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182-183:1–31, 2012.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. URL `http://www.jstatsoft.org/v47/i11/`.

Y. Kameya. Time Series Discretization via MDL-based Histogram Density Estimation. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pages 732–739. IEEE, 2011.

N. Kilbertus, G. Parascandolo, and B. Schölkopf. Generalization in anti-causal learning. *arXiv preprint arXiv:1812.00524*, 2018.

J. Klemelä. Multivariate histograms with data-dependent partitions. *Statistica sinica*, pages 159–176, 2009.

A. Kolmogorov. Three Approaches to the Quantitative Definition of Information. *Problemy Peredachi Informatsii*, 1(1):3–11, 1965.

P. Kontkanen and P. Myllymäki. MDL histogram density estimation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 219–226. JMLR, 2007.

L. F. Kozachenko and N. N. Leonenko. Sample Estimate of the Entropy of a Random Vector. *Problemy Peredachi Informatsii*, 23:9–16, 1987.

L. G. Kraft. *A device for quantizing, grouping, and coding amplitude-modulated pulses.* PhD thesis, Massachusetts Institute of Technology, 1949.

A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

J. Lemeire and E. Dirkx. Causal models as minimal descriptions of multivariate systems, 2006.

J. Lemeire, S. Meganck, F. Cartella, and T. Liu. Conservative independence-based causal structure learning in absence of adjacency faithfulness. *International Journal of Approximate Reasoning*, 53(9):1305–1325, 2012.

M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*, volume 4. Springer, 2019.

F. Liu and L. Chan. Causal Inference on Discrete Data via Estimating Distance Correlations. *Neural Computation*, 28(5):801–814, 2016.

P. Mandros, M. Boley, and J. Vreeken. Discovering Reliable Approximate Functional Dependencies. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 355–364. ACM, 2017.

P. Mandros, D. Kaltenpoth, M. Boley, and J. Vreeken. Discovering Functional Dependencies from Mixed-Type Data. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1404–1414, 2020.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems*, pages 505–511, 2000.

A. Marx and J. Vreeken. Telling Cause from Effect using MDL-based Local and Global Regression. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 307–316. IEEE, 2017.

A. Marx and J. Vreeken. Causal Inference on Multivariate and Mixed-Type Data. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 655–671. IEEE, Springer, 2018.

A. Marx and J. Vreeken. Testing Conditional Independence on Discrete Data using Stochastic Complexity. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019a.

A. Marx and J. Vreeken. Telling cause from effect by local and global regression. *Knowledge and Information Systems*, 60(3):1277–1305, 2019b.

A. Marx and J. Vreeken. Identifiability of Cause and Effect using Regularized Regression. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2019c.

A. Marx, C. Backes, E. Meese, H.-P. Lenhof, and A. Keller. EDISON-WMW: Exact Dynamic Programming Solution of the Wilcoxon-Mann-Whitney Test. *Genomics, Proteomics & Bioinformatics*, 2016.

A. Marx, A. Gretton, and J. M. Mooij. A Weaker Faithfulness Assumption based on Triple Interactions. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI, 2021a.

A. Marx, L. Yang, and M. van Leeuwen. Estimating Conditional Mutual Information for Discrete-Continuous Mixtures using Multidimensional Adaptive Histograms. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*. SIAM, 2021b.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 403–410. AUAI, 1995a.

C. Meek. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–419. AUAI, 1995b.

O. C. Mesner and C. R. Shalizi. Conditional Mutual Information Estimation for Mixed, Discrete and Continuous, Data. *IEEE Transactions on Information Theory*, 2020.

O. Mian, A. Marx, and J. Vreeken. Discovering Fully Oriented Causal Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence.* AAAI, 2021.

T. Mononen and P. Myllymäki. Computing the Multinomial Stochastic Complexity in Sub-Linear Time. In *Proceedings of the International Conference on Probabilistic Graphical Models*, pages 209–216, 2008.

J. Mooij, O. Stegle, D. Janzing, K. Zhang, and B. Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. *Advances in Neural Information Processing Systems*, pages 1687–1695, 2010.

J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

L. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

L. Paninski and M. Yajima. Undersmoothed kernel entropy estimators. *IEEE Transactions on Information Theory*, 54(9):4384–4388, 2008.

J. Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

J. Peters, D. Janzing, and B. Schölkopf. Causal Inference on Discrete Data using Additive Noise Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2436–2450, 2011a.

J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Identifiability of causal graphs using functional Models. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 589–598. AUAI, 2011b.

J. Peters, J. Mooij, D. Janzing, and B. Schölkopf. Causal Discovery with Continuous Additive Noise Models. *Journal of Machine Learning Research*, 15:2009–2053, 2014.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

D. N. Politis. *On the entropy of a mixture distribution*. Purdue University. Department of Statistics, 1991.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 0885-6125.

A. Rahimzamani, H. Asnani, P. Viswanath, and S. Kannan. Estimators for multivariate information measures in general probability spaces. In *Advances in Neural Information Processing Systems*, pages 8664–8675, 2018.

J. Ramsey, P. Spirtes, and J. Zhang. Adjacency-Faithfulness and Conservative Causal Inference. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 401–408. AUAI, 2006.

G. Raskutti and C. Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1), 2018.

H. Reichenbach. *The direction of time*, volume 65. Univ of California Press, 1956.

D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting Novel Associations in Large Data Sets. *Science*, 334(6062):1518–1524, 2011.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(1):465–471, 1978.

J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Technology*, 42(1):40–47, 1996.

J. Rissanen. Strong optimality of the normalized ml models as universal codes and information in data. *IEEE Transactions on Information Theory*, 47(5): 1712–1717, 2001.

W. Rudin. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 (2):461–464, 1978.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons, 2015.

M. Scutari and J.-B. Denis. *Bayesian Networks with Examples in R.* Chapman and Hall, Boca Raton, 2014.

D. Sejdinovic, A. Gretton, and W. Bergsma. A Kernel Test for Three-Variable Interactions. In *Advances in Neural Information Processing Systems*, pages 1124–1132, 2013.

E. Sgouritsa, D. Janzing, P. Hennig, and B. Schölkopf. Inference of Cause and Effect with Unsupervised Inverse Regression. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 38: 847–855, 2015.

C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A Linear Non-Gaussian Acyclic Model for Causal Discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

Y. M. Shtarkov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987.

T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki. Factorized Normalized Maximum Likelihood Criterion for Learning Bayesian Network Structures. *Proceedings of the 4th European Workshop on Probabilistic Graphical Models*, pages 257–264, 2008.

T. Silander, J. Leppä-aho, E. Jääsaari, and T. Roos. Quotient Normalized Maximum Likelihood Criterion for Learning Bayesian Network Structures. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84, pages 948–957. PMLR, 2018.

P. Spirtes and J. Zhang. A Uniformly Consistent Estimator of Causal Effects under the k-Triangle-Faithfulness Assumption. *Statistical Science*, 29(4): 662–678, 2014.

P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richardson. *Causation, prediction, and search.* MIT press, 2000.

W. Spohn. Stochastic Independence, Causal Independence, and Shieldability. *Journal of Philosophical logic*, 9(1):73–99, 1980.

E. V. Strobl, K. Zhang, and S. Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 7(1), 2019.

J. Suzuki. An estimator of mutual information and its application to independence testing. *Entropy*, 18(4):109, 2016.

N. Tagasovska, V. Chavez-Demoulin, and T. Vatter. Distinguishing Cause from Effect Using Quantiles: Bivariate Quantile Causal Discovery. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 119, pages 9311–9323. PMLR, 2020.

C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the Faithfulness Assumption in Causal Inference. *The Annals of Statistics*, pages 436–463, 2013.

G. Valiant and P. Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 685–694, 2011.

J. A. Van Vechten. Quantum dielectric theory of electronegativity in covalent systems. I. Electronic dielectric constant. *Physical Review*, 182(3):891, 1969.

N. Vereshchagin and P. Vitányi. Kolmogorov's Structure functions and model selection. *IEEE Transactions on Information Technology*, 50(12):3265– 3290, 2004.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 220–227. AUAI, 1991.

N. X. Vinh, J. Chan, and J. Bailey. Reconsidering mutual information based feature selection: A statistical significance view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

J. Vreeken. Causal Inference by Direction of Information. In *Proceedings of the SIAM International Conference on Data Mining (SDM), Vancouver, Canada*, pages 909–917. SIAM, 2015.

C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(1):185–194, 1968.

D. Weiler and J. Eggert. Multi-dimensional histogram-based image segmentation. In *International Conference on Neural Information Processing*, pages 963–972. Springer, 2007.

F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

L. Yang, M. Baratchi, and M. van Leeuwen. Unsupervised Discretization by Two-dimensional MDL-based Histogram. *arXiv preprint arXiv:2006.01893*, 2020.

Zhalama, J. Zhang, F. Eberhardt, and W. Mayer. SAT-Based Causal Discovery under Weaker Assumptions. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI).* AUAI, 2017.

J. Zhang. A Comparison of Three Occam's Razors for Markovian Causal Models. *The British journal for the philosophy of science*, 64(2):423–448, 2013.

J. Zhang and P. Spirtes. Detection of Unfaithfulness and Robust Causal Inference. *Minds and Machines*, 18(2):239–271, 2008.

J. Zhang and P. Spirtes. The three faces of faithfulness. *Synthese*, 193(4): 1011–1027, 2016.

K. Zhang and A. Hyvärinen. On the Identifiability of the Post-nonlinear Causal Model. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 647–655. AUAI, 2009.

Y. Zhang, Z. Zhang, K. Liu, and G. Qian. An improved IAMB algorithm for Markov blanket discovery. *Journal of Computers*, 5(11):1755–1761, 2010.

Index