Causal Inference on Discrete Data

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

Kailash Budhathoki

Saarbrücken, 2020

Abstract

Causal inference is one of the fundamental problems in science. To make absolute statements about cause and effect, carefully designed experiments are necessary, in which we consider representative populations, instrument the putative cause, and control for everything else. In practice, setting up such an experiment is often impossible, too expensive, or unethical. The only option then is to consider causal inference from observational studies where data has not been obtained in a controlled manner.

A particularly interesting setting is to tell cause from effect between a pair of random variables X and Y given a sample from their joint distribution. For a long period of time, it was thought to be impossible to distinguish between causal structures $X \to Y$ and $Y \to X$ from observational data as the factorisation of the joint distribution is the same in both directions. In the past decade, researchers have made a long stride in this direction by exploiting sophisticated properties of the joint distribution. Most of the existing methods, however, are for continuous real-valued data.

In the first part of the thesis, we consider bivariate causal inference on different discrete data settings——univariate i.i.d., univariate non-i.i.d., and multivariate i.i.d. pairs. To this end, we build upon the principle of algorithmic independence of conditionals (AIC), which states that marginal distribution of the cause is *algorithmically* independent of conditional distribution of the effect given the cause. However, as Kolmogorov complexity is not computable, we approximate the AIC from above through the statistically sound Minimum Description Length (MDL) principle. On univariate i.i.d. and non-i.i.d. pairs, where causal mechanisms are simple, we use refined MDL codes that are minimax optimal w.r.t. a model class. We resort to crude MDL codes on a pair of multivariate i.i.d. variables.

Although useful, saying that there exists a causal relationship from a set of variables towards a certain variable of interest does not always fully satisfy one's curiosity; for a domain expert it is of particular interest to know those conditions that are most effective, such as the combinations of drugs and their dosages that are most effective towards recovery. Motivated by this problem, in the second part of this thesis, we consider discovering statistically reliable causal *rules* from observational data. Overall, extensive evaluations show that methods proposed in this thesis are highly accurate, and discover meaningful causations from real-world data.

Zusammenfassung

Kausale Inferenz ist eines der grundlegenden Probleme in der Wissenschaft. Um absolute Aussagen über Ursache und Wirkung zu treffen sind sorgfältig geplante Experimente notwendig, in denen wir repräsentative Populationen betrachten, die mutmaßliche Ursache messen und alle weiteren Umstände kontrollieren. In der Praxis ist die Einrichtung eines solches Experiments oft unmöglich, zu teuer oder unethisch. Die einzige Möglichkeit besteht dann darin kausale Schlussfolgerungen aus unkontrollierten Beobachtungsstudien zu ziehen.

Ein Problem von besonderem Interesse ist die Unterscheidung zwishchen Ursache und Wirkung von einem Paar von Zufallsvariablen X and Y. Lange Zeit wurde angenommen dass es unmöglich ist zwischen den kausalen Strukturen $X \to Y$ und $Y \to X$ auf Grundlage von Beobachtungsdaten zu unterscheiden, da die Faktorisierung der gemeinsamen Verteilung in beide Richtungen die gleiche ist. Im letzten Jahrzehnt haben Forscher jedoch große Fortschritte in diesem Gebiet gemacht, indem Sie Komplexe Eigenschaften der gemeinsamen Verteilung geschickt ausnutzen. Die meisten bestehenden Methoden beziehen sich jedoch auf kontinuierliche, reellwertige Daten.

Im ersten Teil der Arbeit, betrachten wir bivariate kausale Inferenz auf verschiedenen diskreten datenquellen—univariat i.i.d., univariat nicht-i.i.d. und multivariat i.i.d. Paare. Zu diesem Zweck bauen wir auf dem Prinzip der algorithmischen Unabhängigkeit von Konditionalen (AIC) auf, das besagt, dass die Randverteilung der Ursache algorithmisch unabhängig von der bedingten Verteilung der Wirkung gegeben der Ursache ist. Da die Kolmogorow-Komplexität jedoch nicht berechenbar ist, nähern wir uns der AIC mithilfe des statistisch soliden Minimum Description Length (MDL) Prinzips von oben an. Bei univariat i.i.d. und nicht i.i.d. Paaren, wo kausale Mechanismen einfach sind, verwenden wir "refined" MDL Kodierungen welche minimax optimal bzgl. einer Modellklasse sind. Wir greifen auf "crude" MDL Kodierungen für multivariat i.i.d. Variablen Paare zurück.

Obwohl die Aussage, dass ein kausaler Zusammenhang von einer Menge von Variablen gegenüber einer bestimmten Variable existiert nützlich ist, erfüllt nicht es immer vollständig jedermanns Interesse. Für Experten in einem Bereich ist es von besonderem Interesse die effektivsten Bedingungen zu kennen, wie z.B. die Kombination und Dosierung von Medikamenten, die für die Genesung am effektivsten sind. Durch diese Problemstellung motiviert, betrachten wir im zweiten Teil dieser Arbeit die Entdeckung kausaler Regeln aus Beobachtungsdaten. Allgemein zeigen umfangreiche Evaluierungen dass die vorgestellten Methoden in dieser Arbeit sehr genau sind und sinnvolle Ursache-Wirkungsverhaltnisse gefunden werden.

Contents

Introduction 1

The ultimate goal of scientific data analysis is to understand the data generating process. Towards that goal, we often study correlations between variables as these may gain us insight into how and which variables interact with each other. For example, identifying how various genes interact with each other is key to understanding the development of tissues and organisms. There exist a large number of techniques at our disposal to that end. With supervised learning, for instance, we can learn a prediction function to understand how genes interact with the phenotype. Often times, not only do we want to find out if variables are correlated, but also guide further actions and policies from there. For example, we might want to deactivate certain genes to treat an illness without side-effects.

Correlations that we observe among variables, however, can mislead us to take wrong actions. A classic example that illustrates this problem is that of the correlation between ice cream sales and violent crimes. In many cities, as ice cream sales rise, so do violent crimes. Based on this observation, should we then stop selling ice cream to reduce violent crimes? Of course not. Based on common sense, we can all agree that ice cream sales go up in warm weather, and violent crimes are also more likely when the temperatures rise. Therefore the correlation that we observe between ice cream sales and violent crimes is due to a third variable *weather*. This simple example shows that to successfully enact on the system, i.e. to decide whether to stop selling ice cream to reduce violent crimes, we have to know more than just the fact that two or more variables are correlated. What we really need to know is their *causal relationship*, i.e. whether one variable *causes* the other.

To answer whether a variable X causes another variable Y, we require a controlled experiment. In such an experiment, we externally change the value of X only and observe Y, while leaving all other variables in the system unchanged. Any change that we observe in Y, therefore, has to be due to X only, and hence we can tell whether X causes Y. Conducting such an experiment in practice, however, is not straightforward. For example, it is extremely difficult to control all factors that potentially influence the recovery (age, sex, diet, environment to name a few) when assessing the efficacy of a drug. A pragmatic alternative then is to conduct a randomised controlled trial where we randomise the assignment of X to individuals, by which we mitigate any influence on Y from factors other than X. In many cases, however, conducting a randomised controlled trial may be simply impossible

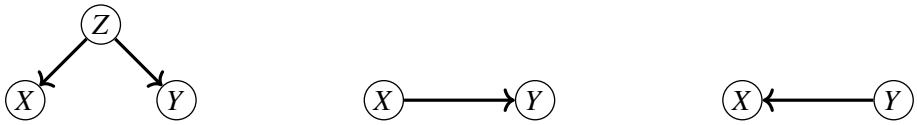


FIGURE 1.1: Reichenbach's common cause principle connects statistical relation to causal relation. A statistical dependence between two variables X and Y implies that either (left) they have a common cause Z, otherwise known as **confounder**, or that (middle) X causes Y, or that (right) Y causes X. Furthermore X and Y are statistically independent when conditioned upon Z. Note that the latter two cases are special cases of the first case when Z coincides with either X or Y.

or at least impractical or unethical. Finding a large number of volunteers willing to face a violent crime—to find out if an increase in violent crime causes an increase in ice cream sales—alone is a daunting task, let alone the ethical side of committing crimes in the name of science.

For a long period of time, it was thought to be impossible to answer causal questions without a controlled experiment. In many cases, we would like to draw causal conclusions from settings where we do not have full control over variables, but are reduced to mere observers, i.e. observational studies. From observational data, we can estimate the joint distribution to understand relationships between variables. However, it may or may not be the same as the joint distribution we would have observed had we run an experiment. Therefore we cannot be certain whether X causes Y using effect measures based only on the joint distribution. The goal of causal inference from observational data is then to identify those conditions under which we can reason about the true causal relationships based on the joint distribution.

Towards this goal, we can start by assessing whether variables are dependent; after all we have been doing that for centuries. However, a well-known axiom in statistics—correlation does not imply causation—suggests that statistical properties of variables alone cannot determine their causal structure. As discussed before, just because ice cream sales correlates with violent crimes does not mean that ice cream sales cause violent crimes. Is it then futile to study dependence between variables in the hope of identifying their underlying causal relation? No. Although we may not identify the exact causal relation, we can infer the *existence* of causal relation between variables from their statistical properties. Reichenbach (1956) was the first one to see this connection.

Principle 1 (Reichenbach's Common Cause Principle)**.** *If two random variables X and Y are statistically dependent ($X \not\perp\!\!\!\perp Y$), then either X causes Y, or that Y causes X, or X and Y have a common cause Z. Moreover, Z screens off the dependence between X and Y in the sense that X and Y become independent ($X \perp\!\!\!\perp Y$) when conditioned upon Z.*

In Figure 1.1, we illustrate this principle with causal graphs where a directed edge from X to Y implies X causes Y. Statistically, $X \to Y$ represents a joint distribution $P(X,Y)$ where we first generate a value of X using the marginal distribution $P(X)$, and then we generate a value of Y using the conditional distribution $P(Y \mid X)$. Let us examine our ice cream sales example through the lens of this principle. We *observe* that ice cream sales (X) and violent crimes (Y) are dependent, and based on *common sense*, we can rule out $X \to Y$ and $Y \to X$. This means that according to this principle, $X \leftarrow Z \to Y$ has to be the explanation for the

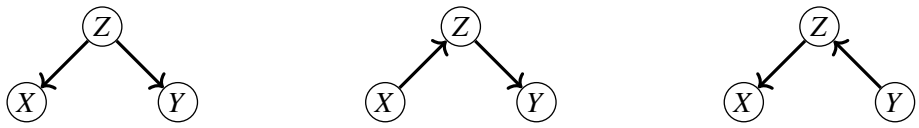


FIGURE 1.2: Markov equivalent causal graphs. The three structures encode the same conditional independence relation: $(X \perp\!\!\!\perp Y) \mid Z$.

dependence between X and Y, with Z for example being the weather or some other unseen factor. Although useful in its own right, this does not solve the problem of causal inference. In scientific practice, we would like to draw conclusions based on the evidence in the form of *observed data* instead of common sense which may not be obvious in many problems.

The big problem with identifying causal graphs from observational data is that different graphs can satisfy the same conditional independence relation. All three causal graphs shown in Figure 1.2, for instance, encode the same conditional independence: $(X \perp\!\!\!\perp Y) \mid Z$. Technically we say they are Markov equivalent. Using Reichenbach's common cause principle, we can hence identify causal structures only up to Markov equivalence classes. Let us consider the implications assuming that we have data over X, Y and Z. Even if we have three variables, and establish that X and Y are independent given Z, we can only infer the undirected graph X—Z—Y. While closer to the truth, we still cannot tell the causal direction between the pairs of dependent variables. The only conclusion we can draw is that *one* of the structures in Figure 1.2 is true, but not which one. At the heart of this problem lies the fact that we cannot distinguish between bivariate causal structures $X \to Y$ and $Y \to X$ because they are also observationally equivalent: $P(X,Y) = P(X)P(Y \mid X) = P(Y)P(X \mid Y)$. For this reason, inferring the causal direction between two variables from their joint distribution was thought to be impossible for a long time. With additional assumptions in the generative model, in the past decade, researchers have shown that bivariate causal inference is possible, however. The general idea is to define classes of marginal and conditional distributions, and choose that direction as the causal direction in which factorisation of the joint distribution belongs to those corresponding classes, but not in the reverse direction.

Most of the existing methods for bivariate causal inference are for continuous real-valued data (Shimizu et al., 2006; Hoyer et al., 2009; Janzing et al., 2012; Bloebaum et al., 2018). In many real-world scenarios, we have data from discrete domain. For example, a person's education level can be finitely many, so is the fact that whether they earn more than 50,000 euros annually. One might then be interested in finding out whether education level causes high income (>50K), or that high income causes education. Those are the kind of problems we consider first. Roughly stated, in the first part of this thesis, we investigate various techniques to infer the bivariate causal structure from observational discrete data. This, we state formally in the research question below:

Question 1. *Given a sample drawn from the joint distribution of two dependent discrete r.v.s X and Y, how do we reliably infer whether $X \to Y$ or $Y \to X$ is their causal structure?*

To answer this question, we build upon the algorithmic independence of conditionals (AIC) principle (Peters et al., 2017b; Janzing and Schölkopf, 2010), which states that the marginal distribution of cause is *algorithmically* independent of the conditional distribution of effect given cause. However, as Kolmogorov complexity (Kolmogorov, 1965) is not computable, we approximate the AIC through the Minimum Description Length (MDL)

principle (Rissanen, 1978). The MDL principle is attractive for various reasons. It provides a statistically sound means for approximating Kolmogorov complexity. Moreover, as the MDL principle considers the trade-off between goodness-of-fit of a model and its simplicity, it naturally avoids overfitting data.

Using the MDL principle, in Chapter 3, we instantiate the AIC through the minimax optimal code with respect to a parametric family of multinomial distributions for a pair of univariate discrete random variables. Building upon the foundations of Granger causality (Granger, 1969), which introduces an additional restriction on the AIC, we use the minimax optimal predictive codes to infer the causal direction between a pair of univariate event sequences—discrete time series—in Chapter 4. In Chapter 5, we instantiate the AIC for a pair of multivariate binary random variables using MDL-based decision trees.

Often times, Question 1 and its answer thereof does not fully satisfy one's curiosity. We want more than just an answer that a variable or a group of variables likely cause a target variable of interest. In combination treatment of drugs, for instance, certain combinations of drugs and their dosages are much more effective than others, while some may even lead to severe side effects. To take any concrete action, it is not enough to know that certain combinations of drugs cause recovery; we have to also know which combination of those drugs and dosages are most effective. For example, in the combination treatment of Tuberculosis (Y), drugs such as Isoniazid (X_1), Pyrazinamide (X_2) and Ethambutol (X_3) are included. Although any off-the-shelf causal inference method might tell us that X_1, X_2 and X_3 cause Y, only certain combinations of their daily dosages (in milligrams per kilogram weight of a patient) are known to be effective, e.g. $X_1 = 5$ `and` $18.2 \leq X_2 \leq 26.3$ `and` $14.5 \leq X_3$, some can also be lethal. This motivates the problem we deal in the second part of this thesis. That is, we would like to find most effective causal *rules* from observational data. This, we state in the research question below.

Question 2. *How do we efficiently discover statistically reliable causal rules from observational data?*

We answer this question for discrete variables in Chapter 6. To this end, we condition our effect measure on a set of potential confounders. This ensures that we measure the causal effect of a rule. Moreover, we give a graphical criteria for all rules that we discover from observational data to be causal. For rules discovered from data to be statistically reliable, we take a conservative approach, and bias our effect measure. The resulting effect measure has a low variance, and therefore rules we discover from sample generalise well to the population. To efficiently search for those rules in data, we propose a branch-and-bound search algorithm. We round up with conclusion in Chapter 7.

Publications

This thesis is a cumulative dissertation based on the research articles shown in Table 1.1. Although many parts of those research articles are included verbatim in this thesis, some parts were rewritten to reflect on the work in hindsight. Moreover, to keep this thesis coherent, we removed abstracts, changed notation, and rewrote introductions from the research articles.

Table 1.1: Publications on which this thesis is based.

publication	used in
K. Budhathoki and J. Vreeken. *MDL for causal inference on discrete data*. In 2017 IEEE 17th International Conference on Data Mining (ICDM)	Chap. 3
K. Budhathoki and J. Vreeken. *Accurate causal inference on discrete data*. In 2018 IEEE 18th International Conference on Data Mining (ICDM)	Chap. 3
K. Budhathoki and J. Vreeken. *Causal inference by compression*. In 2016 IEEE 16th International Conference on Data Mining (ICDM)	Chap. 4
K. Budhathoki and J. Vreeken. *Origo: causal inference by compression*. Knowledge and Information System, Vol. 56, No. 2	Chap. 4
K. Budhathoki and J. Vreeken. *Causal inference on event sequences*. In Proceedings of the 2018 SIAM International Conference on Data Mining (SDM)	Chap. 5
K. Budhathoki, M. Boley, and J. Vreeken. *Rule discovery for exploratory causal reasoning*. In NeurIPS 2018 workshop on Causal Learning (full article under submission)	Chap. 6

Assumptions for Causal Inference 2

In this chapter, we lay out the mathematical foundations for causal inference. Causal analysis aims to infer probabilities under *changing conditions* through interventions. This contrasts with statistical analysis that can only deal with *static* conditions. That is, standard statistical machinery allows us to estimate the joint distribution from a sample by implicitly assuming that external conditions do not change. The joint distribution, however, cannot tell us how it would behave if external conditions were subject to a change. There is nothing in the joint distribution of ice cream sales and violent crime rate alone to tell us that changing ice cream sales would increase or decrease violent crimes. Such information must be provided by causal assumptions which identify relationships that remain the same even when external conditions change. Therefore, there is some causal assumption behind every causal conclusion. A crucial assumption for bivariate causal inference is that of independence of mechanisms.

2.1 The Principle of Independent Mechanisms

Suppose that we observe a sample drawn from the joint distribution $P(X,Y)$. We represent the data-generation process by a causal graph. A causal graph can simulate any data generating process that operates sequentially along it arrows, e.g. a directed acyclic graph. A causal graph such as $X \to Y$ represents a data-generating process where we randomly assign a value to X according to its distribution $P(X)$, and then Y is assigned a value according to the conditional distribution $P(Y \mid X)$.

In bivariate causal inference, our goal is to infer from the sample if $X \to Y$ or $Y \to X$ is a plausible causal graph. That is, we would like to infer whether X causes Y, or Y causes X. We use Pearl's *do*-notation (Pearl, 2009, Chap. 3) $do(X = x)$, or $do(x)$ in short, to represent the **intervention** on X which changes the system by externally forcing X to assume a value of x, keeping everything else in the system fixed. The **effect of an intervention** $do(x)$ on Y is given by the post-intervention distribution $P(Y \mid do(x))$. We say that X causes Y, if changing an intervention on X has a different effect on Y, i.e. $P(Y \mid do(x)) \neq P(Y \mid do(x'))$.

For exposition, let X be the pressure on the acceleration pedal of a car, and Y be the speedometer reading. If we change the pressure on the acceleration pedal, the speedometer

reading will also change; $P(Y \mid do(x)) \neq P(Y \mid do(x'))$. The action of altering the speedometer reading by moving the pointer, however, does not affect the pressure on the acceleration pedal; $P(X \mid do(y))$ does not change, and remains $P(X)$, regardless of the value of Y we set. Therefore the pressure on the acceleration pedal of a car causes its speedometer reading, and not the other way around.

In an observational data, however, we do not have access to the post-intervention distribution $P(Y \mid do(x))$. What we have, instead, is a sample drawn from the joint distribution $P(X,Y)$—and from that sample we can estimate the conditional distribution $P(Y \mid X = x)$. The conditional distribution $P(Y \mid X = x)$, however, can be different than the post-intervention distribution $P(Y \mid do(x))$; whereas everything else in the system is fixed in case of $P(Y \mid do(x))$, that is not necessarily the case for $P(Y \mid X = x)$. It is easy to see this for our car example in the reverse direction from Y to X. Suppose that we have joint observations of X and Y that are representative of the population. The observed conditional distribution of pressure on the acceleration pedal given the value of speedometer reading $P(X \mid Y = y)$ is most likely a unimodal distribution centred around y with some random measurement error. In contrast, $P(X \mid do(Y = y))$ does not depend on the value of Y, and is simply $P(X)$. This problem of identifying the bivariate causal graph from the joint distribution is further complicated by the fact that factorisations of the joint distribution for $X \to Y$ and $Y \to X$ are equivalent, i.e. $P(X)P(Y \mid X) = P(Y)P(X \mid Y)$. Therefore, for a long time, it was believed that we cannot identify the causal graph of two variables using their joint distribution.

If, however, we analyse the assumption hidden in an intervention, there are clues to recovering the causal graph from observational data. In particular, an intervention on a set of variables given their causal graph assumes **modularity** or **invariance**, i.e. in case of a system that consists of two variables, if $X \to Y$ is the underlying causal graph, the conditional distribution of Y given X, $P(Y \mid X)$, does not change if we intervene on X, as long as we do not intervene on Y itself (Property 7.3 Dawid, 2010; Pearl, 2009, Chap. 1.3.2). In other words, no matter what mechanism $P(X)$ we use to regulate X, the conditional distribution $P(Y \mid X)$ remains invariant.

Suppose that we apply pressure on the acceleration pedal depending on a random number generator $P(X)$. No matter what random number generator $P(X)$ we choose to regulate the pressure on the acceleration pedal, the **physical mechanism** $P(Y \mid X)$ responsible for rendering the speedometer reading based on the pressure on the acceleration pedal will remain invariant. In the reverse direction, suppose that we can, somehow, change the speedometer reading by moving the pointer based on a random number generator $P(Y)$. If such a reverse mechanism $P(X \mid Y)$ can be constructed, only specific choices of $P(Y)$ will be able to reproduce $P(X,Y)$ through the factorisation $P(Y)P(X \mid Y)$, as the reverse mechanism is a rather contrived one.

The notion of invariance was formalised as a postulate on the independence of mechanisms particularly for bivariate causal inference by Janzing and Schölkopf (2010), and later stated as a principle in Peters et al. (2017a).

Principle 2 (Independent Mechanisms)**.** *If $X \to Y$ is the underlying causal graph of r.v.s X and Y, then $P(Y \mid X)$ is independent of $P(X)$.*

The notion of **dependence**, however, is abstract. Accordingly, different formalisations have been proposed. IGCI (Janzing et al., 2012) defines dependence in terms of information geometry. Liu and Chan (2016) measure dependence between empirical distributions by

distance correlation. Janzing and Schölkopf (2010) formalise dependence using algorithmic information theory, and postulate algorithmic independence of $P(X)$ and $P(Y \mid X)$. As any physical process can be simulated on a Turing machine (Deutsch, 1985), algorithmic dependence can capture any dependence that can be explained with a physical process; hence the algorithmic causal inference framework is both general and theoretically sound.

2.2 The Algorithmic Independence of Conditionals

We briefly introduce Kolmogorov complexity—a key concept in algorithmic information theory—before formulating the principle of independent mechanisms in terms of algorithmic information theory.

Kolmogorov complexity (Kolmogorov, 1965; Solomonoff, 1964; Chaitin, 1969) measures the complexity of describing a mathematical object, such as numbers, sets, functions, relations. Let $\ell(x)$ denote the length of a binary string x. Let $\mathscr{U}(p)$ denote the output of the universal Turing machine $\mathscr{U}$ for an input program p. The **Kolmogorov complexity** of a finite binary string x is denoted by $K(x)$, and defined as the length of the shortest binary program to $\mathscr{U}$ that generates x and stops. Formally, we have

$$K(x) = \min_{p:\mathscr{U}(p)=x} \ell(x) \, .$$

We can think of the shortest program p^* as the most succinct algorithmic description of x, and $K(x)$ as the ultimate lossless compressed size of x.

The Kolmogorov complexity of a probability distribution P, denoted $K(P)$, is the length of the shortest program to $\mathscr{U}$ that outputs $P(x)$ to a precision ε on the input $\langle x, \varepsilon \rangle$ (Grünwald and Vitányi, 2008). The **conditional Kolmogorov complexity** of a probability distribution P, given a probability distribution Q, is denoted by $K(P \mid Q)$, and defined as the length of the shortest binary program to $\mathscr{U}$ that outputs $P(x)$ to a precision ε on the input $\langle x, \varepsilon, Q \rangle$. We can then define the **algorithmic mutual information** between P and Q as

$$I(P : Q) = K(P) - K(P \mid Q^*) \stackrel{+}{=} K(Q) - K(Q \mid P^*) \, ,$$

where P^* and Q^* are the lengths of the shortest binary program for P and Q respectively, and $\stackrel{+}{=}$ indicates that the equality holds up to an additive constant. Using algorithmic information theory, we can now formalise the principle of independent mechanisms.

Principle 3 (Algorithmic Independence of Conditionals (Peters et al., 2017a))**.** *If $X \to Y$ is the underlying causal graph of random variables X and Y, then $P(Y \mid X)$ is algorithmically independent of $P(X)$, i.e.*

$$I(P(X) : P(Y \mid X)) \stackrel{+}{=} 0,$$

or equivalently $K(P(X)) + K(P(Y \mid X)) \stackrel{+}{\leq} K(P(Y)) + K(P(X \mid Y))$.

The algorithmic independence of conditionals (AIC) implies that the factorisation of the joint distribution $P(X,Y)$ is simpler—in terms of Kolmogorov complexity—in the causal direction than in the anti-causal direction. Thus, we can identify the underlying causal graph by comparing the Kolmogorov complexity of the factorisation of $P(X,Y)$ between two directions. Leaving aside the computability of Kolmogorov complexity, causal inference

using the AIC requires access to the joint distribution $P(X,Y)$. In practice, we only have a sample drawn from $P(X,Y)$, but not $P(X,Y)$ itself.

In practice, we have to not only estimate the probability densities from the sample, we have to also compute their Kolmogorov complexity. With a large enough sample size, and appropriate smoothness, we can get a fairly good estimate of the probability densities through kernel density estimation. What we cannot do, however, is compute their Kolmogorov complexity, amongst others due to the halting problem. In practice, we therefore need other, computable, notions of independence or information. In Chapter 3—5, we instantiate the algorithmic independence of conditionals with the Minimum Description Length principle.

2.3 Structural Equation Model

Structural equation modeling is, arguably, the most popular framework for causal analysis. In a structural equation model (SEM), we represent data generating process by a set of structural assignments (Pearl, 2009, Chap. 1.4). Given a set of variables and their causal graph, an SEM represents every variable as a deterministic function of its parents in the causal graph and an unobserved noise variable. In case of two variables X and Y with the underlying causal graph $X \to Y$, an SEM consists of two assignments:

$$\begin{aligned} X &:= f_X(\emptyset, N_X) \,, \\ Y &:= f_Y(X, N_Y) \,, \end{aligned}$$

where f_X and f_Y are deterministic functions, and noise variables N_X and N_Y are statistically independent, i.e. $N_Y \perp\!\!\!\perp N_Y$.

Note that we use an assignment operator ($:=$) instead of an equality operator ($=$) to indicate a functional dependence in an SEM. Unlike the algebraic equality operator which allows us to move variables on both sides of the equation, the assignment operator does not. The assignment operator has a causal meaning in an SEM; any intervention on X leads to a change in Y. In an SEM, an intervention such as $do(x)$ can be carried out by replacing the corresponding structural assignment by $X := x$. As such, after the intervention $do(x)$, the SEM reduces to

$$\begin{aligned} X &:= x \,, \\ Y &:= f_Y(X, N_Y) \,. \end{aligned}$$

The distribution of Y entailed by this modified SEM corresponds to the post-intervention distribution $P(Y \mid do(x))$.

So far we assumed that we have access to the causal graph. Our goal is instead to identify the causal graph itself. One may then wonder whether we can identify the causal graph from the joint distribution with SEMs. For general SEMs, without restrictions on the functional form or the distribution of noise, we can always construct a suitable function and noise in both directions such that noises are jointly independent (Peters et al., 2017b, Prop. 4.1). Therefore, we cannot tell if the joint distribution $P(X,Y)$ is induced by an SEM with the causal graph $X \to Y$, or $Y \to X$. That is, we cannot **identify** causal graph from the joint distribution with general SEMs. With some restrictions, however, we can recover the causal graph from the joint distribution.

Next we discuss a special class of SEMs, known as Additive Noise Models, that do allow us to identify the causal graph from the joint distribution.

2.4 Additive Noise Model (ANM)

Given a set of variables and their causal graph, an Additive Noise Model (ANM) represents every variable as a deterministic function of its parents and an additive noise variable (Pearl, 2009, Chap. 7.1.2). In case of two variables X and Y with the underlying causal graph $X \to Y$, an ANM consists of two assignments:

$$\begin{aligned} X &:= N_X \ , \\ Y &:= f_Y(X) + N_Y \ , \end{aligned}$$

where $N_Y \perp\!\!\!\perp N_X$, or equivalently $N_Y \perp\!\!\!\perp X$. If the data-generation process follows an ANM, we can identify the causal structure from the joint distribution if variables admit an ANM in one direction, but not in the other. That is, in the anti-causal direction, we cannot fit a function and an additive noise such that noise variables are independent.

The assumption that noises are independent can also be seen as an instance of the principle of independent mechanisms. In case of two variables with the causal graph $X \to Y$, for instance, the marginal distribution of cause $P(X)$ is the same as the distribution of its noise $P(N_X)$, and the conditional $P(Y \mid X)$ is the same as the marginal $P(N_Y)$. Thus assuming $N_X \perp\!\!\!\perp N_Y$ is equivalent to assuming $P(X) \perp\!\!\!\perp P(Y \mid X)$.

In the past decade, we have gained an extensive understanding on the identifiability of causal graphs from the joint distribution using ANMs. Shimizu et al. (2006) showed that we can distinguish causal directions for linear models with non-Gaussian additive noise. A rather less extreme restriction on the class of ANMs also allows for causal inference in practice. The identifiability result from Hoyer et al. (2009) implies that a joint distribution "generally" does not admit an ANM in both directions at the same time. As such, as long as the function is non-linear, there are no restrictions on the distribution of noise for the causal direction to be identifiable. Peters et al. (2010) extend ANMs to discrete variables, and show that in "general" a joint distribution admits an ANM in *at most* one direction.

In practice, to identify the causal direction given a sample drawn from the joint distribution, we first fit ANMs in both directions and choose the direction with the independence as the causal direction. In particular, in the fitting step, we find the pair (f_Y, N_Y) in the direction from X to Y and the pair (f_X, N_X) in the reverse direction. Then, with a suitable independence test, we check whether $N_Y \perp\!\!\!\perp X$, or $N_X \perp\!\!\!\perp Y$ holds. If the results of the independence tests are the same in both direction, we cannot identify the structure using ANM—we are undecided. Otherwise we pick the direction with the independence as the causal direction. As a result, causal inference using ANM hinges on the choice of independence measure.

Most dependence measures either assume the type of the sampling distribution of the test statistic, or require a kernel. Alternatively information-theory offers Shannon entropy (Cover and Thomas, 2006) as an intuitive yet powerful tool to measure independence without classical hypothesis testing based on p-values. In Chapter 3, we show how to instantiate ANMs through both information theory and algorithmic information theory.

Bivariate Causal Inference on IID Data 3

In this chapter, we consider the problem of inferring causal direction between a pair i.i.d. univariate discrete random variables from a sample drawn from their joint distribution.[1]

3.1 Introduction

Suppose that we have observations of two variables: a student's grade and socio-economic status of the student's family—both of which can take on only a finite number of values. Many studies report that these are statistically dependent (Hill and Giammatteo, 1963; Croizet and Dutrévis, 2004). How can we then tell if one is the cause of the other? Intuitively, changing socio-economic status of a student's family will most likely affect the student's grade, whereas changing a student's grade would probably not affect socio-economic status of their family. It is therefore plausible that socio-economic status of a student's family causes student's grade, and not the other way around. We do not have experimental data, however, to answer the question—what we have instead is observational data.

In a nutshell, we would like to infer the causal direction between a pair of i.i.d. univariate discrete random variables from a sample drawn from their joint distribution. Most of the existing methods for bivariate causal inference (Shimizu et al., 2006; Mooij et al., 2009; Janzing et al., 2012; Peters et al., 2014; Bloebaum et al., 2018) work only with continuous real-valued data. Although there exist a few bivariate causal inference methods for discrete data (Peters et al., 2010; Liu and Chan, 2016; Kocaoglu et al., 2017; Cai et al., 2018), they require accurate estimation of the distribution from the sample for reliable causal inference. Commonly used plug-in estimators are known to overfit and hence inference at the population level from small sample sizes can be unreliable.

To avoid overfitting, thereby generalise results at the population level, we can instead turn to the Minimum Description Length (MDL) principle (Rissanen, 1978; Grünwald, 2007). As the MDL principle provides a statistically sound means for approximating Kolmogorov complexity (Vereshchagin and Vitanyi, 2004; Gács et al., 2001), this allows us to formulate a computable version of the algorithmic independence of conditionals (AIC). Moreover,

[1] This chapter builds upon and extends Budhathoki and Vreeken (2017, 2018a).

using a MDL-based estimator, we can draw a connection between the AIC and other causal inference frameworks, such as additive noise models (ANMs) (Peters et al., 2010).

In this work, we present the computable (algorithmic)-information-theoretic formulations of existing bivariate causal inference frameworks. First we show how to instantiate the AIC for discrete data using the refined version of MDL. Then we provide an information-theoretic formulation of ANMs using Shannon entropy as a dependence measure—as such we avoid explicit statistical hypothesis testing. Lastly we use an MDL-based estimator of Shannon entropy within an ANM. The information-theoretic formulation gives us general, efficient, identifiable, and, as the experiments show, highly accurate methods for bivariate causal inference from a sample of discrete variables.

3.2 Refined MDL-based Approximation of AIC

Consider a pair of correlated univariate discrete random variables X and Y with their finite domains $\mathscr{X}$ and $\mathscr{Y}$ respectively. Suppose that we have a sample drawn from their joint distribution $P(X,Y)$. From this sample, we would like to infer whether X causes Y, or Y causes X. To this end, first we use the refined version of the Minimum Description Length principle (MDL) (Rissanen, 1978) to approximate the algorithmic independence of conditionals (AIC).

3.2.1 Refined MDL: Stochastic Complexity (SC)

The practical version of the Minimum Description Length (MDL) principle (Rissanen, 1978) provides a statistically sound means for approximating Komogorov complexity (Gács et al., 2001; Vereshchagin and Vitanyi, 2004). Rather than all programs to the Turing machines, it only considers those for which we know they stop after generating the desired output, i.e. lossless compressors. The MDL principle is particularly suitable for materialising the algorithmic independence of conditionals, as its generic solution to the model selection problem demands density estimation from the sample go hand in hand with the complexity of the estimated density.

In MDL literature, programs are often referred to as models. Typically models are a family of probability distributions or functions with the same functional form, e.g. a parametric family of Poisson distributions, k^{th} degree polynomials. Informally, according to the MDL principle, just like the two-part decomposition of Kolmogorov complexity, the best model to explain the data D from a set of models (model class) $\mathscr{M}$ is the one that minimises $L(M)+L(D \mid M)$, where $L(M)$ is the description length, in bits, of the model M, and $L(D \mid M)$ is the length, in bits, of the data when encoded with the model M (Grünwald, 2007).

Given M, we can encode D using the optimal prefix code of length $L(D \mid M) = -\log P(D \mid M)$, where $P(D \mid M)$ is the probability mass or density of D w.r.t. M. To encode M, we have to make a choice out of many possible codes (coding schemes), which leaves room for arbitrariness; $L(P)$ can be large under one code, but relatively shorter under another. The *refined* version of MDL overcomes this arbitrariness by encoding data D not just with one model M, but with the entire model class $\mathscr{M}$. It is possible to design a one-part code of length $\bar{L}(D \mid \mathscr{M})$ for any data D such that it differs from the shortest code length of

D using individual models in $\mathscr{M}$ by a constant. Codes with such property are also called Universal codes.

One such code can be obtained by constructing a Normalized Maximum Likelihood (NML) distribution from the model class, and taking the optimal prefix code of data according to its NML distribution (Rissanen, 2000). Suppose that our model class is a parametric family of distributions,[2] i.e. $\mathscr{M} = \{P(\bullet \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$, where $\Theta \subseteq \mathbb{R}^m$ is a m-dimensional parameter space. Let $x^n = (x_i)_{i=1}^n$ be a sequence of n outcomes where each outcome x_i is an element of $\mathscr{X}$. Let $\mathscr{X}^n$ be the n-fold Cartesian product of $\mathscr{X}$ such that $x^n \in \mathscr{X}^n$. Then the NML distribution w.r.t. a model class $\mathscr{M}$ is defined as

$$P_{\mathrm{NML}}(x^n;\mathscr{M}) = \frac{P(x^n \mid \hat{\boldsymbol{\theta}}(x^n;\mathscr{M}))}{\sum_{z^n \in \mathscr{X}^n} P(z^n \mid \hat{\boldsymbol{\theta}}(z^n;\mathscr{M}))}\,, \tag{3.1}$$

where $\hat{\boldsymbol{\theta}}(x^n;\mathscr{M})$ is the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ w.r.t. $\mathscr{M}$ for x^n. As the name suggests, the NML distribution of x^n relative to a model class $\mathscr{M}$ is the maximum likelihood of x^n w.r.t. to $\mathscr{M}$ normalised over the sum of the maximum likelihoods of every possible sample of size n w.r.t. to $\mathscr{M}$. The NML distribution has a number of important theoretical properties. First, it gives a unique solution to the minimax problem posed by Shtarkov (1987),

$$\min_{\bar{P}} \max_{x^n \in \mathscr{X}^n} \log \frac{P(x^n \mid \hat{\boldsymbol{\theta}}(x^n,\mathscr{M}))}{\bar{P}(x^n \mid \mathscr{M})}\,.$$

That is, for *any* data x^n from $\mathscr{X}^n$, $P_{\mathrm{NML}}(x^n;\mathscr{M})$ assigns a probability, which differs from the highest achievable probability within the model class—the maximum likelihood $P(x^n \mid \hat{\boldsymbol{\theta}}(x^n;\mathscr{M}))$— by a constant factor in the denominator of Equation (3.1). In other words, the NML distribution is the minimax optimal universal model with respect to the model class. Second, it also provides solution to another minimax problem formulated by Rissanen (2001), which is given by

$$\min_{\bar{P}} \max_{Q} \mathbb{E}_Q \left(\log \frac{P(x^n \mid \hat{\boldsymbol{\theta}}(x^n;\mathscr{M}))}{\bar{P}(x^n;\mathscr{M})} \right)\,,$$

where Q is the worst-case data generating distribution (outside the model class $\mathscr{M}$), and $\mathbb{E}_Q$ is the expectation over x^n. That is, even if the true data generating distribution does *not* reside in the model class $\mathscr{M}$ under consideration, $P_{\mathrm{NML}}(x^n \mid \mathscr{M})$ *still* gives the optimal encoding for the data x^n relative to $\mathscr{M}$. These properties are highly desirable when modelling real-world problems, where we often do not know the true distribution, yet want our model to perform as close to the true model as possible, regardless of whether the true model lies inside or outside the model class.

The optimal prefix code length of x^n corresponding to its NML distribution w.r.t. $\mathscr{M}$ is also called the **stochastic complexity** of x^n w.r.t. $\mathscr{M}$, defined as

$$\begin{aligned} S(x^n;\mathscr{M}) &= -\log P_{\mathrm{NML}}(x^n;\mathscr{M}) \\ &= -\log P(x^n \mid \hat{\boldsymbol{\theta}}(x^n;\mathscr{M})) + \log \sum_{z^n \in \mathscr{X}^n} P(z^n \mid \hat{\boldsymbol{\theta}}(z^n;\mathscr{M}))\,, \end{aligned}$$

where the last term with the summation is the **parametric complexity** of the model class $\mathscr{M}$. Given that we only have data x^n and a contemplated model class $\mathscr{M}$, which may or may not contain the true distribution, it is plausible to instantiate $K(P(X))$ through $S(x^n;\mathscr{M})$.

[2] We can convert deterministic functions into distributions by adding a random noise.

3.2.2 Stochastic Complexity for Multinomials

Let $x^n = (x_i)_{i=1}^n$ be a sequence of n outcomes where each outcome x_i is an element of $\mathscr{X} = \{1, 2, \ldots, m\}$. For discrete random variables, we consider a family of multinomial distributions as our contemplated model class, which is defined as

$$\mathscr{M}_m = \{P(X \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta_m\} ,$$

where Θ_m is the simplex shape parameter space given by

$$\Theta_m = \{\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m) \mid \theta_j \geq 0 \text{ and } \sum_{j=1}^m \theta_j = 1\} ,$$

and $\theta_j = P(X = j \mid \boldsymbol{\theta})$. The MLE of $\boldsymbol{\theta}$ w.r.t. $\mathscr{M}_m$ from the sample x^n is given by

$$\hat{\boldsymbol{\theta}}(x^n; \mathscr{M}_m) = \left(\frac{h_1}{n}, \frac{h_2}{n}, \ldots, \frac{h_m}{n} \right) ,$$

where h_j is the number of occurrences (frequency) of an outcome j in sample x^n, i.e.

$$h_j = |\{x_i \mid x_i = j, i = 1, 2, \ldots, n\}| .$$

Thus the NML distribution of x^n w.r.t. the multinomial model class $\mathscr{M}_m$ is given by

$$P_{\mathrm{NML}}(x^n; \mathscr{M}_m) = \frac{\prod_{j=1}^m (h_j/n)^{h_j}}{\sum_{h_1 + \cdots + h_m = n} \binom{n}{h_1, \ldots, h_m} \prod_{j=1}^m (h_j/n)^{h_j}} ,$$

where the multinomial coefficient is the number of ordered arrangements of outcomes of x^n such that each outcome j occurs h_j times. Then the stochastic complexity of x^n w.r.t. $\mathscr{M}_m$ is

$$\begin{aligned} S(x^n; \mathscr{M}_m) &= -\log \prod_{j=1}^m (h_j/n)^{h_j} + \log \sum_{h_1 + \cdots + h_m = n} \binom{n}{h_1, \ldots, h_m} \prod_{j=1}^m (h_j/n)^{h_j} \\ &= n H_n(X) + \log R(\mathscr{M}_m, n) , \end{aligned} \tag{3.2}$$

where $H_n(X)$ is the plug-in estimate of Shannon entropy (Cover and Thomas, 2006) of X, using the empirical distribution $\hat{P}(\bullet)$ on the sample x^n, and $\log R(\mathscr{M}_m, n)$ is the parametric complexity of the multinomial model class with m distinct outcomes and the sample size of n.

Computational Complexity We can compute the counts h_js in $O(n)$ time with a single pass over the data. Although the parametric complexity is exponential in m, we can approximate it up to a finite floating-point precision of d digits in sub-linear time with respect to the sample size n given precomputed counts h_i (Mononen and Myllymäki, 2008).[3] Altogether we can compute the multinomial stochastic complexity of x^n in $O(n)$ time.

[3] In the experiments, we set $d = 10$.

3.2.3 Conditional SC for Multinomials

Let $x^n = (x_i)_{i=1}^n$ be a sequence of n outcomes where each outcome x_i is an element of the space of outcomes $\mathscr{X}$ of size $|\mathscr{X}| = m$. Let $y^n = (y_i)_{i=1}^n$ be another sequence of n outcomes where each outcome y_i is an element of $\mathscr{Y}$ whose size is $|\mathscr{Y}| = \ell$. To compute the conditional stochastic complexity of y^n given x^n w.r.t. the multinomial model class, we use the optimal prefix code of y^n given x^n according to its factorized Normalized Maximum Likelihood distribution (fNML) (Silander et al., 2008). The fNML distribution of y^n given x^n, denoted $P_{\text{fNML}}(y^n \mid x^n; \mathscr{M}_\ell)$, is the product of the NML probabilities of the parts of y^n that are defined by the values of x^n, which is given by

$$P_{\text{fNML}}(y^n \mid x^n; \mathscr{M}_\ell) = \prod_{x \in \mathscr{X}} P_{\text{NML}}(y^n \mid x; \mathscr{M}_\ell) \ .$$

Thus the stochastic complexity of y^n given x^n using fNML is given by

$$\begin{aligned} S(y^n \mid x^n; \mathscr{M}_\ell) &= -\log P_{\text{fNML}}(y^n \mid x^n; \mathscr{M}_\ell) \\ &= -\log \prod_{x \in \mathscr{X}} P_{\text{NML}}(y^n \mid x; \mathscr{M}_\ell) \\ &= \sum_{x \in \mathscr{X}} -\log P_{\text{NML}}(y^n \mid x; \mathscr{M}_\ell) \\ &= \sum_{x \in \mathscr{X}} S(y^n \mid x; \mathscr{M}_\ell) \ . \end{aligned}$$

That is, we can compute the conditional stochastic complexity of y^n given x^n by partitioning y^n according to the unique values of x^n, computing the multinomial stochastic complexity of each part of y^n, and finally aggregating them.

Computational Complexity On a single simultaneous pass over x^n and y^n, we can partition y^n as well as compute the counts in each part. We can compute the multinomial stochastic complexity of each part of size $n^{(x)}$ in $O(n^{(x)})$ time. Thus, to compute the multinomial stochastic complexities of all parts of y^n, it takes $\sum_{x \in \mathscr{X}} O(n^{(x)}) \equiv O(n)$ time. Altogether we can compute the conditional multinomial stochastic complexity in $O(n)$ time.

3.2.4 Multinomial SC based AIC for Discrete Data

Using the marginal and the conditional stochastic complexity for multinomials, we can now state the stochastic complexity based approximation of AIC for discrete data.

Proposition 3.2.1 (Multinomial Stochastic Complexity based approx. of AIC)**.** *Consider a sample x^n of the discrete random variable X with its sample space $\mathscr{X}$ of size m. Likewise let y^n be a sample of the discrete random variable Y with its sample space $\mathscr{Y}$ of size ℓ. If $X \to Y$ is the underlying causal graph, then*

$$S(x^n; \mathscr{M}_m) + S(y^n \mid x^n; \mathscr{M}_\ell) < S(y^n; \mathscr{M}_\ell) + S(x^n \mid y^n; \mathscr{M}_m) \ .$$

Using this proposition, we can identify the causal direction from the sample as follows:

- Infer $X \to Y$ if $S(x^n; \mathscr{M}_m) + S(y^n \mid x^n; \mathscr{M}_\ell) < S(y^n; \mathscr{M}_\ell) + S(x^n \mid y^n; \mathscr{M}_m)$.
- Infer $Y \to X$ if $S(y^n; \mathscr{M}_\ell) + S(x^n \mid y^n; \mathscr{M}_m) < S(x^n; \mathscr{M}_m) + S(y^n \mid x^n; \mathscr{M}_\ell)$.

- Undecided otherwise.

We refer to this causal inference procedure as CISC. Although the formulation above follows directly from the statistically sound approximation of Kolmogorov complexity provided by the MDL principle, there are no theoretical guarantees why the above proposition should hold. After all, we are approximating the Kolmogorov complexity of a distribution by the stochastic complexity of data w.r.t. a parametric model class. In short, CISC lacks theoretical results on whether we can identify the true causal graph from the joint distribution of a pair of variables. For bivariate causal inference, there are other practically successful sound theoretical frameworks that possess the identifiability that we seek. Next we discuss how we can formulate one such framework—additive noise models (ANMs) (Pearl, 2009, Chap. 1)—in terms of information theory.

3.2.5 Information-Theoretic ANM

To arrive at the information-theoretic formulation of ANMs, we have to quantify the information content—in terms of joint Shannon entropy—of random variables X and Y with the joint distribution $P(X,Y)$ induced by ANMs in two directions: $X \to Y$ and $Y \to X$. Thus, in addition to computing the marginal Shannon entropies of X and Y, we need the conditional Shannon entropy of Y given X and vice versa. Although trivial, the lemma below implies that the conditional Shannon entropy of Y given X is the same as the conditional Shannon entropy of noise N_Y given X in an ANM.

Lemma 3.2.1. *For an ANM from X to Y, we have $H(Y \mid X) = H(N_Y \mid X)$.*

Proof. The conditional Shannon entropy of Y given X is defined as

$$\begin{aligned} H(Y \mid X) &= \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x) \\ &= \sum_{x \in \mathcal{X}} P(X = x) H(f_Y(x) + N_Y \mid X = x) \end{aligned}$$

as adding a constant $f_Y(x)$ to a random variable N_Y only changes the outcomes of N_Y, but not its distribution, we have $H(f_Y(x) + N_Y \mid X = x) = H(N_Y \mid X = x)$, which results in

$$\begin{aligned} &= \sum_{x \in \mathcal{X}} P(X = x) H(N_Y \mid X = x) \\ &= H(N_Y \mid X) \,. \end{aligned}$$

□

With this lemma, we can now present the information-theoretic formulation of ANMs.

Theorem 3.2.2 (Information-Theoretic Additive Noise Model)**.** *Consider the joint distribution $P(X,Y)$ induced by an ANM with the causal graph $X \to Y$. Thus in the "generic" case, there exists a function f_Y such that $N_Y = Y - f_Y(X)$ is independent of X, but for any function f_X, $N_X = X - f_X(Y)$ depends on Y (Peters et al., 2010). Then for a function f_Y and any function f_X, we have*

$$H(X) + H(N_Y) < H(Y) + H(N_X) \,.$$

Proof. The joint Shannon entropy of random variables X and Y whose joint distribution is induced by an ANM with the causal graph $X \to Y$ is given by

$$\begin{aligned} H(X) + H(Y \mid X) &= H(X) + H(N_Y \mid X) && \text{(using Lemma 3.2.1)} \\ &= H(X) + H(N_Y)\,. && (\because N_Y \perp\!\!\!\perp X) \end{aligned}$$

In the other direction $Y \to X$, we have the joint Shannon entropy of X and Y as

$$\begin{aligned} H(Y) + H(X \mid Y) &= H(Y) + H(N_X \mid Y) && \text{(using Lemma 3.2.1)} \\ &< H(Y) + H(N_X)\,. && (\because N_X \not\perp\!\!\!\perp Y) \end{aligned}$$

Since the joint Shannon entropy of X and Y is symmetric in their ordering, we get the final inequality by combining the right hand sides of the relations above. □

This theorem shows that we can infer the causal direction by simply comparing the Shannon entropy of random variables under ANMs in two directions.[4] In practice, however, we do not have access to the true distribution. Instead, we have to estimate Shannon entropy of a random variable from a sample. To this end, we can start by using the naive plug-in estimator of Shannon entropy based on the empirical distribution. Let $H_n(X)$ be the plug-in estimate of Shannon entropy of the random variable X from the sample x^n using the empirical distribution. Using H_n as an estimator of H in Theorem 3.2.2, we can identify the causal direction from the sample as follows:

- Infer $X \to Y$ if $H_n(X) + H_n(N_Y) < H_n(Y) + H_n(N_X)$.
- Infer $Y \to X$ if $H_n(Y) + H_n(N_X) < H_n(X) + H_n(N_Y)$.
- Undecided otherwise.

We refer to this causal inference procedure as ACID. Using Shannon entropy, we avoid explicit statistical hypothesis testing for independence. Moreover calculating Shannon entropy is computationally cheaper than running an independence test, such as the chi-squared test of independence. The plug-in estimators are, however, known to overfit data. An alternative is to consider multinomial stochastic complexity (see Chap. 3.2.2) which can also be seen as an estimator of Shannon entropy. Although biased, causal inference using multinomial stochastic complexity generalise better to the population. Next we discuss how to use multinomial stochastic complexity as an estimator of Shannon entropy with ANMs.

3.2.6 Multinomial Stochastic Complexity based ANM

Based on multinomial stochastic complexity, we propose the following estimator for Shannon entropy.

$$S_n(x^n; \mathscr{M}_m) = S(x^n; \mathscr{M}_m)/n\,.$$

The following theorem shows that $S_n(x^n; \mathscr{M}_m)$ is strongly consistent for every possible probability distribution.

[4] For continuous real-valued data, Kpotufe et al. (2014) studied similar formulations with differential entropy.

Theorem 3.2.3. *The multinomial stochastic complexity based estimate of Shannon entropy is strongly universally consistent, that is, almost surely*

$$\lim_{n\to\infty} S_n(x^n, \mathscr{M}_m) = H(X) \ .$$

Proof. Using Eq. (3.2), we can express $S(x^n; \mathscr{M}_m)$ as

$$S(x^n; \mathscr{M}_m) = nH_n(X) + \log R(\mathscr{M}_m, n) \ .$$

Using this, we can write $S_n(x^n, \mathscr{M}_m)$ as

$$S_n(x^n, \mathscr{M}_m) = H_n(X) + \frac{\log R(\mathscr{M}_m, n)}{n} \ .$$

Thus to prove the theorem, we have to show that

$$\lim_{n\to\infty} \left(H_n(X) + \frac{\log R(\mathscr{M}_m, n)}{n} \right) = H(X) \ .$$

To this end, first we simplify the left hand side of the above expression, this results in

$$\lim_{n\to\infty} \left(H_n(X) + \frac{\log R(\mathscr{M}_m, n)}{n} \right) = \lim_{n\to\infty} H_n(X) + \lim_{n\to\infty} \frac{\log R(\mathscr{M}_m, n)}{n} \ .$$

From the result by Antos and Kontoyiannis (2001), we know that the plug-in estimator of Shannon entropy is strongly universally consistent, that is,

$$\lim_{n\to\infty} H_n(X) = H(X) \ . \tag{3.3}$$

To show that the second limit with parametric complexity vanishes, we use the asymptotic expansion of the parametric complexity (Rissanen, 2000), i.e.

$$\log R(\mathscr{M}_m, n) = \frac{m}{2} \log \frac{n}{2\pi} + \log \frac{\pi^{(m+1)/2}}{\Gamma(\frac{m+1}{2})} + o(1) \ ,$$

where $\Gamma(\bullet)$ is the Euler gamma function and $o(1) \to 0$ as $n \to \infty$. Note that we can upper bound the logarithm of a number by its square root, i.e. $0 \leq \log n \leq \sqrt{n}$ for all $n > 0$. Thus we have $\lim_{n\to\infty} \frac{\log n}{n} \leq \lim_{n\to\infty} \frac{\sqrt{n}}{n} = \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0$, and trivially $\lim_{n\to\infty} 0 = 0$. Applying the squeeze theorem, we get $\lim_{n\to\infty} \frac{\log n}{n} = 0$. With this, it is easy to see that

$$\lim_{n\to\infty} \frac{\log R(\mathscr{M}_m, n)}{n} = 0 \ . \tag{3.4}$$

Combining relations (3.3) and (3.4), we get the final result. □

Just as in case of the plug-in estimator of Shannon entropy, we can now simply replace H by S_n in Theorem 3.2.2. By fitting a function $\hat{f}_Y$, we estimate the noise as $y^n - \hat{f}_Y(x^n)$ in the direction from X to Y; let α be the size of the domain of the estimated noise. Likewise, by fitting a function $\hat{f}_X$, we estimate the noise as $x^n - \hat{f}_X(y^n)$ in the direction from Y to X; let β be the size of the domain of the estimated noise. Using S_n as an estimator of H in Theorem 3.2.2, we can identify the causal direction from the sample as follows:

- Infer $X \to Y$ if $S(x^n, \mathscr{M}_m) + S(y^n - \hat{f}_Y(x^n), \mathscr{M}_\alpha) < S(y^n, \mathscr{M}_\ell) + S(x^n - \hat{f}_X(y^n), \mathscr{M}_\beta)$.
- Infer $Y \to X$ if $S(y^n, \mathscr{M}_\ell) + S(y^n - \hat{f}_Y(x^n), \mathscr{M}_\beta) < S(x^n, \mathscr{M}_m) + S(x^n - \hat{f}_X(y^n), \mathscr{M}_\alpha)$.
- Undecided otherwise.

Note that as n is on the both side of the inequalities, we omitted n from our causal inference rule, and simply use S instead of S_n in the comparison. We refer to this causal inference procedure as CRISP from here onwards.

3.2.7 Information-Theoretic Discrete Regression

To use ANM-based causal inference rules, we need the noise variables, their distributions in particular. Of course, we do not know the true distributions of the noise variables. By fitting a function on the sample in each direction, however, we can compute the empirical distributions of the noise variables, e.g. in the direction from X to Y, find the function $\hat{f}_Y$ such that we can estimate the noise by $y^n - \hat{f}_Y(x^n)$. This is a well-known regression problem, with discrete variables. The discrete regression algorithm we present here is an adaptation of discrete regression with dependence minimisation (Peters et al., 2010) to information-theoretic scores.

To find the function such as $\hat{f}_Y$ in a typical regression problem, we minimise a loss function, such as the residual sum of squares (RSS) or the ℓ_p norm. Such loss functions, however, are not appropriate for our purpose; after the regression, we check whether the residual is independent of the regressor. Thus we need a loss function that maximises the independence between the residual and the regressor. In the information-theoretic terms, this implies minimising the mutual information between the residual and the regressor, which is equivalent to minimising the Shannon entropy of the residual.

Unlike in the continuous case, there is no risk of overfitting in the discrete case; Y may take different values for each outcomes of X, and hence there is no need for regularization. We can simply consider all possible functions, and take the one with the minimal value of the loss function. However, even if range of the function lies within the domain of the target variable, we are left with exponentially many choices of functions, thereby making the problem intractable. Hence, we resort to heuristics.

We present the pseudocode for information-theoretic discrete regression in Algorithm 1. To regress Y as a function of X, we start with a function that maps each x-value to the most frequently co-occurring y value (line 1-2). Then we iteratively update the function for each x value. To ensure that the algorithm is deterministic, we do so in some canonical order (line 8). To update the function for a x value, we temporarily map x to other y values keeping all other mappings $f(x')$ with $x \neq x'$ fixed. We use $f_{j-1}^{x \to y}(x^n)$ to denote that f_{j-1} temporarily maps x to y. From all the mappings, we pick the best one as the one that results in the least estimated Shannon entropy of the residual (line 9). If the estimated entropy of this residual is better than the best estimated Shannon entropy of the residual so far, we update our function (line 10-13). We keep on iterating as long as the estimated Shannon entropy of the residual reduces, or we arrive at the maximum number of iterations J (line 14).

In a nutshell, we perform alternating minimisation in discrete space. Note that entropy estimators presented here are non-negative, and hence is bounded from below. Since the search space is finite and the estimated Shannon entropy of the residual is strictly decreasing in every iteration, the algorithm will converge. It could, however, converge to

Algorithm 1: Discrete Regression with Entropy Estimate Minimisation

Input: Discrete sequences x^n and y^n, max. no. of iterations J, entropy estimator $\hat{H}$
Output: $\hat{H}(y^n - \hat{f}(x^n))$

$\mathscr{X} \leftarrow \text{SET}(x^n)$;
$\mathscr{Y} \leftarrow \min(y^n), \min(y^n)+1, \ldots, \max(y^n)$;
for $x \leftarrow \mathscr{X}$ **do**
    $\hat{f}_0(x) \leftarrow \arg\max_{y \in \mathscr{Y}} \hat{P}(X = x, Y = y)$;
$e \leftarrow \hat{H}(y^n - \hat{f}_0(x^n))$;
$j \leftarrow 0$;
do
    $j \leftarrow j + 1$;
    for $x \leftarrow \text{RANDOMORDER}(\mathscr{X})$ **do**
        $e' \leftarrow \min_{y \in \mathscr{Y}} \hat{H}(y^n - \hat{f}_{j-1}^{x \to y}(x^n))$;
        if $e' < e$ **then**
            $e \leftarrow e'$;
            $\hat{f}_j(x) \leftarrow \operatorname{argmin}_{y \in \mathscr{Y}} \hat{H}(y^n - \hat{f}_{j-1}^{x \to y}(x^n))$;
while $j < J$;
return $\hat{H}(y^n - f_j(x^n))$

a local optimum. The worst case computational complexity of the discrete regression is $O(|\mathscr{Y}|^{|\mathscr{X}|}) \equiv O(\ell^m)$. By setting the maximum number of iterations J, we can terminate early, however.

3.3 Related Work

Most of the existing methods for causal inference on a pair of discrete variables are either based on the structural equation models (SEMs), or the algorithmic independence of conditionals (AIC).

In SEMs, every variable is assumed to be a deterministic function of its parents and an unobserved noise variable (Pearl, 2009, Chap. 1.4). The additive noise models (ANMs) are a special class of SEMs which assume that the noise is additive. Peters et al. (2010) extend ANMs to discrete data, and propose the DR algorithm. DR uses chi-squared test of independence, which is more expensive to compute than Shannon entropy. Moreover, we do not require explicit null hypothesis testing in every iteration, unlike DR.

Another causal inference method for a pair of discrete variables based on SCMs is ECI (Kocaoglu et al., 2017). They postulate that the unobserved variable (noise) is simpler—in terms of the Rényi entropy—in the true direction. In particular, it is conjectured that if X causes Y, then we have $H_\alpha(X) + H_\alpha(N) < H_\alpha(Y) + H_\alpha(\tilde{N})$ with H_α being the Rényi entropy, where $Y = f(X,N), X \perp\!\!\!\perp N$ and $X = g(Y,\tilde{N}), X \perp\!\!\!\perp \tilde{N}$. Unlike ANMs, which assume that the noise is additive, it can be of arbitrary type in ECI.

The algorithmic independence of conditionals (AIC) postulates that if X causes Y, $P(X)$ and $P(Y \mid X)$ are *algorithmically* independent (Janzing and Schölkopf, 2010; Lemeire and Dirkx, 2006; Peters et al., 2017b). As Kolmogorov complexity is not computable, causal

inference methods based on the AIC have to define a computable dependence measure. Liu and Chan (2016) (DC), for instance, use distance correlation as a dependence measure. To infer the causal direction, DC computes the distance correlation between empirical marginal and conditional distributions in two directions.

A recent proposal by Cai et al. (2018) (HCR) takes a different approach than these two frameworks. They assume a two-stage causal process that consists of a deterministic map from the cause to a hidden compact representation, and a probabilistic map from the hidden representation to the effect. This causal model is identifiable under some conditions on the true causal mechanism.

3.4 Experiments

We implemented all the proposed causal inference methods for discrete variables in Python and provide the source code, along with the used datasets, and synthetic dataset generator.[5] All experiments were executed single threaded on MacBook Pro with 2.5 GHz Intel Core i7 processor and 16 GB memory. Following Peters et al. (2010), we use the statistical significance level of $\alpha = 0.05$ for DR, and set the maximum number of iterations J to 10 for the discrete regression.

3.4.1 Synthetic Data

First we consider synthetic data to study the performance of causal inference methods on data with known ground truth. To this end, we generate synthetic data with the ground truth $X \to Y$ using ANMs. Following the scheme of Peters et al. (2010), we sample cause, i.e. X, from following model classes:

- uniform from $\{2, \ldots, L\}$,
- binomial with parameters (n, p),
- geometric with parameter p,
- hypergeometric with parameters (M, K, N),
- poisson with parameter λ,
- negative binomial with parameter (n, p), and
- multinomial with parameter $\boldsymbol{\theta}$.

We randomly choose the parameters for each model class. In particular, we choose L uniformly between 2 and 10, p uniformly between 0.1 and 0.9, n, M and K uniformly between 1 and 40, N uniformly between 1 and $\min(40, M+K-1)$, λ uniformly between 1 and 10, and $\boldsymbol{\theta}$ randomly s.t. $\sum_j \boldsymbol{\theta}_j = 1.0$. We choose $f(x)$ uniformly between -7 and +7 for every x, and noise N uniformly, independent of X, between $-t$ and $+t$, where t is uniformly chosen between 1 and 7.

All the existing methods (discussed in Chapter 3.3) except DR assume generative models that are not ANMs. It is, therefore, fair to include only DR for comparison. As for the evaluation, the problem of inferring causal direction is similar to that of binary classification. It does not make sense, however, to report ROC curves, which are typically used for

[5] `https://github.com/kailashbuki/caddie`

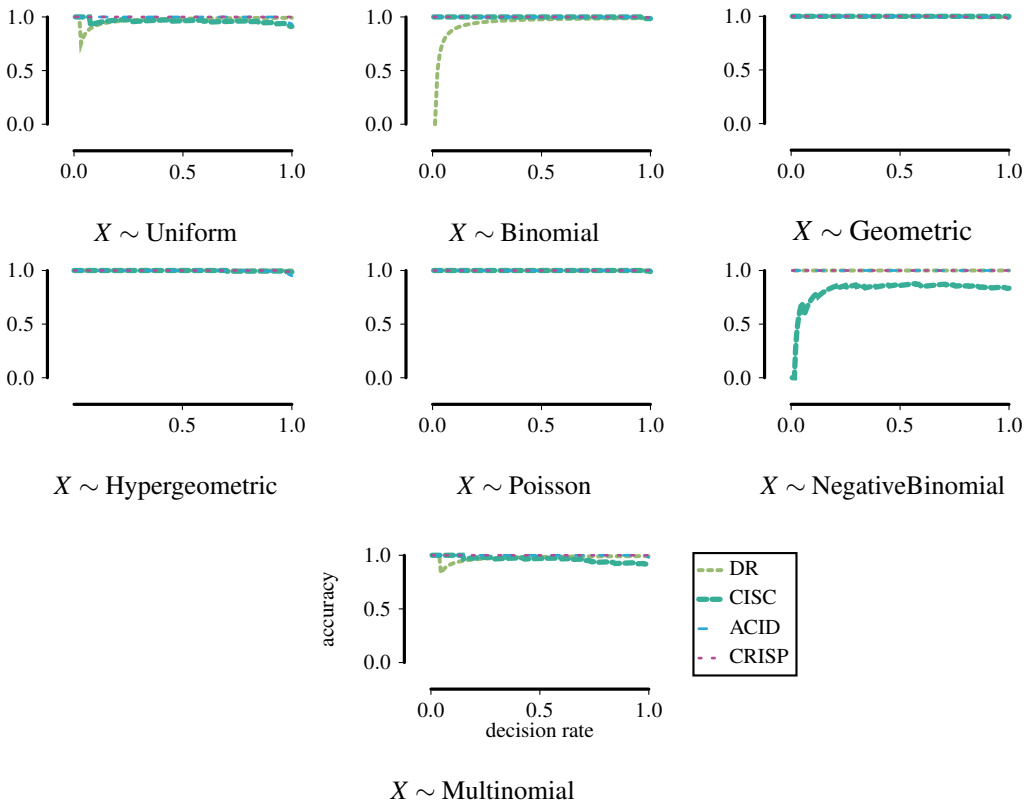


FIGURE 3.1: Accuracy (fraction of correct decisions among decided cause-effect pairs) against decision rate (fraction of samples for which algorithm makes a decision for causal direction) for various distributions of X.

evaluating classifiers. ROC curves enforce asymmetry between positive and negative classes. In our task, however, we do not have such asymmetry—we can change class labels by simply swapping the variables. Therefore accuracy is a more natural metric in our setting. To balance the classes, we swap X and Y variables for a half of the pairs.

We start by assessing the performance of various methods against the difference in their scores in two directions. Let $C_{X\to Y}$ denote the score in the direction from X to Y using a specific method, and $C_{X\to Y}$ be that in the reverse direction using the same method. If we only take decisions for pairs with $|C_{X\to Y} - C_{Y\to X}| \geq \delta$ for some threshold δ, we can trade-off *accuracy* (percentage of correct decisions) versus *decision rate* (percentage of pairs in which a decision was taken). One problem remains that some of the pairs may have $\delta = 0$, therewith the method remains undecided. As a result, we may not achieve a 100% decision rate. To circumvent this problem, we first generate 1000 cause-effect pairs, and only consider those pairs for which $\delta > 0$ for further analysis.

Figure 3.1 shows accuracies of various methods at increasing decision rates for 1000 cause-effect pairs for various model classes. The results show that all methods except CISC are highly accurate whenever they make a decision, in all cases. To complement these results further, we also show *decisiveness* (percentage of decisions among all cause-effect pairs) of causal inference methods for the aforementioned model classes in Figure 3.3. We

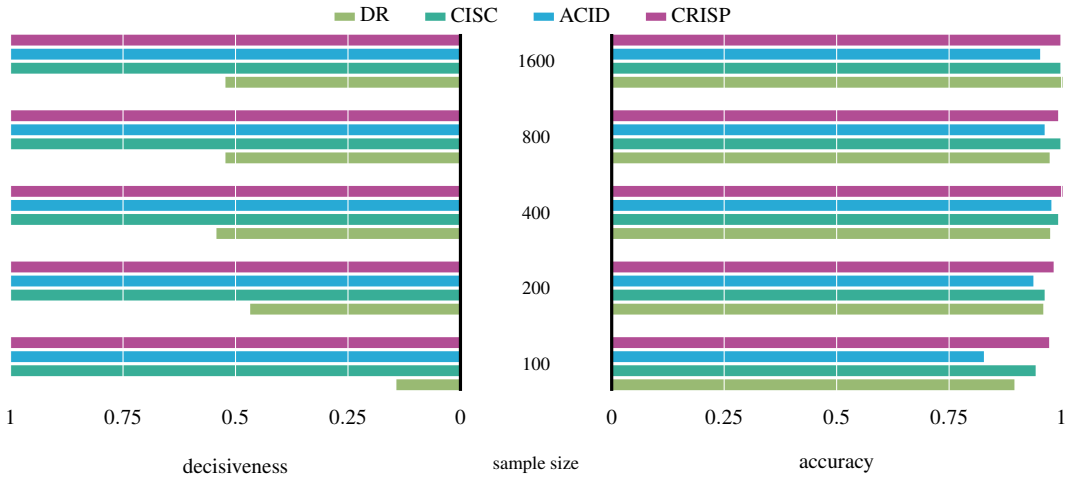


FIGURE 3.2: Accuracy (percentage of correct decisions among decided cause-effect pairs) and decisiveness (percentage of decisions among all cause-effect pairs) against sample size on cause-effect pairs with X generated randomly from the Geometric family.

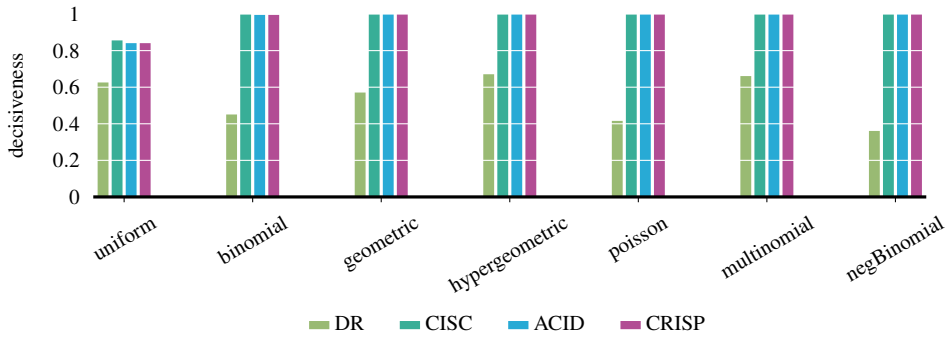


FIGURE 3.3: Decisiveness (percentage of decided pairs) of bivariate discrete causal inference methods for various distributions of X.

observe that DR is often indecisive. The proposed methods, however, make decisions in most cause-effect pairs, regardless of the model class.

Next we study the effect of sample size on the performance of causal inference methods. To this end, we generate 1000 cause-effect pairs from the geometric model class. In Figure 3.2 (left), we show the decisiveness of various methods at various sample sizes. We observe that all methods but DR are highly decisive even when the sample size is small—in the range of hundreds. In Figure 3.2 (right), we show the accuracy on the decided cause-effect pairs. We see that DR is accurate even though it is not as decisive compared to other methods. The proposed methods (CRISP in particular), however, are not only highly decisive, but they are also highly accurate starting from a small sample size.

Overall we observe that information-theoretic causal inference methods on discrete data are both highly decisive and accurate when causal generative model follows an ANM.

Table 3.1: Results on real-world datasets. A tick (✓) indicates a correct decision, a cross (×) indicates a wrong decision, and a double-headed arrow (↔) indicates an indecision.

Dataset	Ground Truth	CISC	DR	ACID	CRISP
abalone	sex → length	×	↔	✓	✓
	sex → diameter	×	↔	✓	✓
	sex → height	×	↔	✓	✓
nlschools	SES → test-score	×	✓	✓	✓

3.4.2 Real-World Data

Abalone The `abalone` dataset from the UCI machine learning repository[6] contains physical measurements of 4 177 abalones (large, edible sea snails). We consider *sex* (X) of the abalone against length (Y_1), diameter (Y_2), and height (Y_3). The sex of the abalone is nominal (male, female, or infant), whereas length, diameter, and height are all measured in millimeters, and have 70, 57 and 28 unique values, respectively. Following Peters et al. (2010), we treat the data as discrete. Since sex causes the size of the abalone and not the other way around, we regard $X \to Y_1$, $X \to Y_2$, and $X \to Y_3$ as the ground truth. We report the results in Table 3.1. Both ACID and CRISP recover the ground truth from all pairs. In contrast, CISC infers wrong direction in all of them. DR, on the other hand, remains undecided in all cases; it is not wrong, however.

NLSchools The `nlschools` dataset is the 99-th pair in the Tübingen cause-effect benchmark pairs.[7] It contains the language test score (X), and socio-economic status (SES) of pupil's family (Y) of 2287 eighth-grade pupils from 132 classes in 131 schools in the Netherlands. The language test score has 47 unique values, and the socio-economic status of pupil's family has 21 unique values. Existing research (Croizet and Dutrévis, 2004; Hill and Giammatteo, 1963) in social science point to $Y \to X$ as the ground truth. Intuitively it also makes sense that the socio-economic status of pupil's family is one of the causes of the language test score. As shown in Table 3.1, all methods but CISC recover the ground truth.

3.5 Discussion

In this work, we showed how to instantiate the algorithmic independence of conditionals (AIC) through the Minimum Description Length (MDL) principle for bivariate causal inference on discrete data. Taking an alternative route, we also gave an information-theoretic formulation of additive noise models (ANMs) using Shannon entropy. Furthermore, we proposed an MDL-based estimator of Shannon entropy within an ANM. The experiments show that the proposed methods are highly accurate on both synthetic and real-world data.

These results suggest that information-theoretic formulation of ANM leads to reliable causal inference from samples. In particular, we do not require evaluating p-values, unlike other statistical independence testing frameworks that require evaluating p-values in every

[6] `http://archive.ics.uci.edu/ml/`

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

iteration. Furthermore we can estimate Shannon entropy—using both plug-in and MDL-based estimators—relatively faster than running hypothesis tests for independence.

Although these results are promising, we see many possibilities for future work. CISC does not have any theoretical guarantees on the identifiability of the data generating mechanism, unlike ACID and CRISP. It is important to note that identifiability results apply only at the population level. Therefore, having an identifiability result ensures that we are estimating the right quantity from a sample using statistical machinery available to us. It would make an engaging future work to include identifiability results for CISC.

In the experiments, the proposed methods, ACID and CRISP in particular, achieve (near) 100% accuracy on both synthetic and real-world data. Although impressive, those results by no means suggest that the proposed methods are perfect. In particular, when domain sizes of variables are much larger than the sample size, performances of all causal inference methods will inevitably drop. We can see a glimpse of this behaviour in Figure 3.2 when the sample size is a mere 100.

3.6 Conclusion

We studied the problem of inferring causal direction between a pair of i.i.d. discrete random variables from a sample drawn from their joint distribution. To this end, first we instantiated the algorithmic independence of conditionals (AIC) for discrete data through the refined version of the Minimum Description Length principle. The resulting inference procedure CISC, however, does not have any theoretical guarantees on whether it identifies the true causal graph. The causal graph of discrete additive noise models (ANMs) are known to be identifiable in the generic case. Taking advantage of this identifiability property, we formulated discrete ANMs in terms of information theory using Shannon entropy as a dependence measure, and proposed ACID. Shannon entropy is cheaper to compute, and we do not require explicit statistical hypothesis testing for independence. However, as the plug-in estimator of Shannon entropy overfits, we proposed CRISP using the refined-MDL-based estimator of Shannon entropy that is biased, but generalises well. Extensive evaluation on synthetic and real-world data shows that the proposed methods are highly accurate on a wide range of settings.

Software Artefacts

The Python implementation of causal inference methods used in this chapter has been released as a python package `caddie` in the PyPI repository.

Installation

The package requires Python $\geq$ 3.7. To install the package and all its dependencies, use `pip3`.

```
$ pip3 install caddie
```

Example Usage

For all the methods, we report results in a tuple of the form $(C_{X\to Y}, C_{Y\to X})$. For information-theoretic methods, we report bits; for methods that employ hypothesis testing, we report the p-value.

```
>>> X, Y = [1, 1, 1, 1, 1], [-1, -1, -1, -1, -1]
>>> from caddie import anm, cisc, measures as mrs
>>> cisc.cisc(X, Y)
>>> anm.fit_both_dir(X, Y, mrs.StochasticComplexity)
>>> anm.fit_both_dir(X, Y, mrs.ChiSquaredTest)
>>> anm.fit_both_dir(X, Y, mrs.ShannonEntropy)
```

Bivariate Causal Inference on Event Sequences 4

In the previous chapter, we considered i.i.d. data. In many real-world applications, we have non-i.i.d. data. Next we consider discrete-valued times series, or event sequences.[1]

4.1 Introduction

Suppose that we have two recording stations along a river. At those recording stations, over a period of time, we measure the water level every 15 minutes. Every time we measure the water level, we can simply record whether the water level increased, decreased or stayed the same (three possible values) compared to the last reading. How can we tell the direction of the river simply by looking at those records? Intuitively water level upstream causes water level downstream, with some lag. As such, the river follows from the upstream to the downstream recording station, and not the other way around. We can agree that this trivial exercise would not be so trivial any more had we not known that the data was from a river.

In a nutshell, we consider the case where we are given two discrete-valued time series—event sequences—of length n, and have to determine whether it is more likely that x^n caused y^n, or that y^n caused x^n. Most of the existing bivariate causal inference methods for time series (Granger, 1969; Rissanen and Wax, 1987; Chen et al., 2004; Chu and Glymour, 2008; Hyvärinen et al., 2008; Peters et al., 2013; Huang and Kleinberg, 2015), however, work with continuous real-valued data. Transfer entropy (Schreiber, 2000) is an information-theoretic variant of Granger causality (Granger, 1969) that is directly applicable to event sequences. However, as it uses the plug-in prediction strategy, it is not robust to model misspecification (Kotlowski and Grünwald, 2012).

In this work we take a related, but subtly different approach. We take an information theoretic viewpoint and define causality in terms of compression. Simply put, we say that x^n causes y^n if we save more bits by compressing the data of y^n with additionally the past of x^n, than vice versa. To optimally compress the data, we would need to know its distribution. In practice, however, we only have observed data and a class of possible prediction strategies—in which the true distribution may or may not live. We hence build

[1] This work is published as Budhathoki and Vreeken (2018b).

our inference framework on the notion of sequential normalized maximum likelihood (SNML) (Kotlowski and Grünwald, 2012), which is a strategy that is guaranteed to give the minimum number of additional bits (regret) compared to the true distribution, regardless of input, and regardless of whether or not the true distribution is in the model class under consideration. At every time step, our prediction for the current outcome is proportional to the Maximum Likelihood estimate of the overall sequence, including the past outcomes as well as the current one.

We propose a bivariate causal inference method on event sequences using SNML, including a detailed exposition on how to derive our causal indicators for binary event sequences based on the class of bernoulli distributions—from which the extension to multinomial distributions is trivial. Importantly, for discrete data in general, CUTE, which stands for **c**a**u**sal inference on even**t** sequenc**e**s, has only a linear time worst case runtime complexity. We empirically evaluate CUTE on a wide range of binary-valued event sequences. Results on synthetic data show that it performs better than transfer entropy on a wide range of settings. Additionally, we consider two case studies on real world data, where we find that CUTE with high accuracy reconstructs the ground truth in water elevation levels in two rivers, as well as in discovering excitatory connections in neural spike train data.

4.2 Theory

In this section, we formally introduce the problem, and present our framework.

4.2.1 The Problem, Formally

Let $x^n = x_1, x_2, \ldots, x_n$ be an event sequence, a time series of n observed outcomes where each outcome x_i is an element of a discrete space of observations $\mathscr{X} \in \{1, 2, \ldots, m\}$. Likewise $y^n = y_1, y_2, \ldots, y_n$ such that $y_i \in \mathscr{Y}$. Given two correlated event sequences x^n and y^n, we are interested in finding the most likely causal direction between them. That is, we would like to identify whether x^n causes y^n, or y^n causes x^n, or they are just correlated.

4.2.2 Assumptions

To measure the causal dependence between two event sequences, we take the usual assumptions of Granger causality (Granger, 1969). Namely, we assume the following.

1. Cause precedes the effect in time.

2. Cause has unique information about the future values of effect.

Assumption 1 is commonly accepted (Chen et al., 2004; Wu and Hatsopoulos, 2006), and also corroborated by the thermodynamic principle—the arrow of causation points in the same direction as the arrow of time. That is, the past influences the future, but not the other way around. One of the implications of Assumption 2 is that we assume there is no confounding event sequence z^n that is the common cause of both x^n and y^n. The other implied assumption is that there is no instantaneous causal relationship—the present value of the cause does not help in the prediction of the present value of the effect. Assumption 2 is also intuitively plausible: the past of the cause and the future of the effect should share some information which cannot be accounted for only by the knowledge of the past of the

effect. This also means that causal dependence measure should be able to quantify that unique information which is not available otherwise.

4.2.3 Measuring Causal Dependence

We base our causal dependence measure on the foundation of Granger causality (Granger, 1969) where causal dependence is measured in terms of *predictability*.

Definition 4.2.1 (Granger Causality)**.** *Let I^t be the information available as of time t in the entire universe that includes both x^{t-1} and y^{t-1}, and $I^t_{\neg x}$ be that in a modified universe where x^{t-1} is excluded. We say that x^t Granger-causes y^t if*

$$P(y_{t+1} \mid I^t) > P(y_{t+1} \mid I^t_{\neg x}) \ ,$$

where P indicates the prediction strategy.[2]

Building upon ideas from Rissanen and Wax (1987), we associate predictability with compression. In particular, we consider the encoded length of the event sequence using a sequential prediction strategy. Intuitively the more predictable an event sequence is, the smaller the number of bits required to describe it using the prediction strategy.

Let $P(x_t \mid x^{t-1})$ be the prediction of current outcome x_t given its past x^{t-1}. To encode the event sequence x^n, we use $P(\bullet \mid x^{t-1})$ in every iteration $t = 1, 2, \ldots, n$. Let $P : \mathscr{X}^n \to [0,1]$ be the probability distribution over all the possible event sequences of size n from $\mathscr{X}$, and $P(\mathscr{X}^n = x^n)$ be the probability mass of the event sequence x^n. Then the predictions $P(\bullet \mid x^{t-1})$ can be considered as a conditional of the joint distribution, i.e. $P(\mathscr{X}^n = x^n) = \prod_{t=1}^n P(x_t \mid x^{t-1})$.

The ideal code length for encoding the current outcome x_t given its past x^{t-1} using the prediction $P(x_t \mid x^{t-1})$ is $-\log P(x_t \mid x^{t-1})$. In learning theory, it is commonly known as *log loss*. Hence the total encoded length of the event sequence x^n using its past, denoted $L(x^n)$, is given by

$$L(x^n) = \sum_{t=1}^{n} -\log P(x_t \mid x^{t-1}) \ .$$

Likewise, let $P(x_t \mid x^{t-1}, y^{t-1})$ be the prediction probability of x_t given the past outcomes of x^n, as well as the past outcomes of y^n. The total encoded length of the event sequence x^n using its past as well as the past of y^n, denoted $L(x^n \mid y^n)$, is then

$$L(x^n \mid y^n) = \sum_{t=1}^{n} -\log P(x_t \mid x^{t-1}, y^{t-1}) \ .$$

Note that the encoded size $L(x^n)$ measures the predictability of x^n from its past outcomes, and $L(x^n \mid y^n)$ measures the predictability of x^n from its past, as well as the past of y^n. Their difference, hence, measures the extra predictability of x^n contributed by the past of y^n which is not available otherwise. With that, we define the causal dependence from the direction y^n to x^n as

$$\Delta_{y^n \to x^n} = L(x^n) - L(x^n \mid y^n) \ ,$$

[2] In the original paper (Granger, 1969), predictability is measured in terms of the variance of the error in regression, thereby ending up with a reverse inequality.

and that from x^n to y^n is given by

$$\Delta_{x^n \to y^n} = L(y^n) - L(y^n \mid x^n) \,.$$

Due to the dependence on time our causal dependence measure is inherently asymmetric. Under our assumptions, the direction with larger dependence is likely the true causal direction. Thus, using the above indicators we arrive at the following causal inference rules on event sequence data.

- If $\Delta_{x^n \to y^n} > \Delta_{y^n \to x^n}$, we infer $x^n \to y^n$.
- If $\Delta_{x^n \to y^n} < \Delta_{y^n \to x^n}$, we infer $y^n \to x^n$.
- Undecided otherwise.

That is, if the added knowledge of the past outcomes of x^n makes the encoding of y^n easier than vice versa, we infer x^n is likely the cause of y^n. If it is the other way around, we infer y^n is likely the cause of x^n. If causal dependence is the same in both directions, we remain undecided. The larger the difference in causal dependence in both directions, the more confident we are. In practice, we can always introduce a threshold τ on the absolute difference between two indicators $|\Delta_{x^n \to y^n} - \Delta_{y^n \to x^n}|$, and treat the results smaller than τ as undecided.

The proposed causal inference rule is based on the premise that we have access to the *true* distribution. In practice, we of course do not know this distribution; we only have *observed* data, and possible models or prediction strategies $\mathscr{P}$. The true distribution *may* or *may not* be in this model class. Next we discuss how to construct a prediction strategy such that we get optimal performance w.r.t. $\mathscr{P}$, regardless of whether true distribution lies in $\mathscr{P}$.

4.2.4 Sequential Normalised Maximum Likelihood

As models, prediction strategies $\mathscr{P}$, we consider parameterised families of distributions. Formally, we define $\mathscr{P}$ as

$$\mathscr{P} = \{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\} \,,$$

where Θ is a parameter space, i.e. $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^k\}$, and $k > 0$. Typically the performance of a prediction strategy P on an event sequence x^n w.r.t. a model class $\mathscr{P}$ is measured by *regret*, which is defined as

$$\begin{aligned} R(P; x^n) &= \sum_{t=1}^{n} -\log P(x_t \,|\, x^{t-1}) - \min_{P_{\boldsymbol{\theta}} \in \mathscr{P}} \sum_{t=1}^{n} -\log P_{\boldsymbol{\theta}}(x_t \,|\, x^{t-1}) \\ &= -\log P(x^n) - \min_{P_{\boldsymbol{\theta}} \in \mathscr{P}} -\log P_{\boldsymbol{\theta}}(x^n) \,. \end{aligned}$$

In words, regret is the additional number of bits required to encode the event sequence using a prediction strategy P instead of the best prediction strategy from the model class $\mathscr{P}$. The regret, however, is not the same for all $x^n \in \mathscr{X}^n$—it can be small for some, and large for others. To be robust against model misspecification, the worst-case regret is taken over all possible event sequences of length n:

$$R_{\max}(P; n) = \max_{x^n \in \mathscr{X}^n} R(P; x^n) \,.$$

The optimal prediction strategy relative to a model class $\mathscr{P}$ for a sample of size n is then the one that minimises the worst-case regret,

$$\min_P R_{\max}(P;n) \ .$$

If the true data generating distribution lies in the model class under consideration $\mathscr{P}$, the maximum likelihood (ML) strategy—predict the next outcome x_{t+1} using the distribution $P_{\hat{\boldsymbol{\theta}}(x^t)}$ with $\hat{\boldsymbol{\theta}}(x^t)$ being the ML estimator based on the past outcomes x^t—will be the optimal prediction strategy. The ML strategy, however, is not robust against the misspecifaction of the model class, i.e. when the true distribution is not in the model class under consideration the result can be arbitrarily bad (Kotlowski and Grünwald, 2012).

We would like to have a prediction strategy P that is optimal regardless of whether the true distribution lies in $\mathscr{P}$. A surprisingly slight modification of the ML strategy can achieve such optimality, and gives the solution to the minimax problem posed above. The modification involves computing the ML estimator of the data sequence including the current outcome, followed by the normalisation of the distribution. That is, the modified strategy predicts x_t with a distribution proportional to $P_{\hat{\boldsymbol{\theta}}(x^{t-1},x_t)}$, where $\hat{\boldsymbol{\theta}}(x^{t-1},x_t)$ is the ML estimator for the data sequence $x_1,\ldots,x_{t-1},x_t$, and is defined as

$$P_{\mathrm{SNML}}(x_t \mid x^{t-1}) = \frac{P_{\hat{\boldsymbol{\theta}}(x^{t-1},x_t)}}{\sum\limits_{x\in\mathscr{X}} P_{\hat{\boldsymbol{\theta}}(x^{t-1},x)}} \ .$$

This strategy is also known as the Sequential Normalised Maximum Likelihood model (SNML) (Kotlowski and Grünwald, 2012; Rissanen and Roos, 2007). We use it to encode the event sequence. For the exponential family of distributions (e.g. Bernoulli, Multinomial, Gaussian, etc.), we can use the respective closed-form expression to calculate the ML estimator $\hat{\boldsymbol{\theta}}$. Hence, it turns out to be easy to compute the SNML strategy for the whole exponential family.

Importantly the SNML strategy is general in the sense that we are only restricted by the choice of our model class. For clarity, we focus specifically on binary data. Without loss of generality, it generalises to the general discrete case.

4.2.5 SNML for Binary Data

As models for binary data, we consider a parameterised family of Bernoulli distributions. The parameterised family of Bernoulli distributions $\mathscr{B}$ is defined as $\mathscr{B} = \{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, where Θ is a parameter space defined as $\Theta = \{\boldsymbol{\theta} \in [0,1]\}$. The probability mass function for Bernoulli distribution is given by

$$P_{\boldsymbol{\theta}}(X = k) = \boldsymbol{\theta}^k(1-\boldsymbol{\theta})^{1-k} \ ,$$

where $k \in \{0,1\}$. The ML estimator for an event sequence x^{t-1} relative to the Bernoulli class is given by $\hat{\boldsymbol{\theta}}(x^{t-1}) = t_1/(t-1)$, where $t_1 = \sum_{i=1}^{t-1} x_i$ is the number of ones in x^{t-1}. Let $t_0 = t-1-t_1$ be the number of zeros in x^{t-1}. Then the denominator of the SNML strategy

for predicting x_t given the past x^{t-1} is given by

$$\begin{aligned}
&\sum_{x\in\mathscr{X}} P_{\hat{\boldsymbol{\theta}}(x^{t-1},x)} = \sum_{x\in\{0,1\}} P_{\hat{\boldsymbol{\theta}}(x^{t-1},x)} = P_{\hat{\boldsymbol{\theta}}(x^{t-1},0)} + P_{\hat{\boldsymbol{\theta}}(x^{t-1},1)} \\
&= (\hat{\boldsymbol{\theta}}(x^{t-1},0))^{t_1}(1-\hat{\boldsymbol{\theta}}(x^t,0))^{t_0+1} + \\
&\quad (\hat{\boldsymbol{\theta}}(x^{t-1},1))^{t_1+1}(1-\hat{\boldsymbol{\theta}}(x^t,1))^{t_0} \\
&= \left(\frac{t_1}{t}\right)^{t_1}\left(1-\frac{t_1}{t}\right)^{t_0+1} + \left(\frac{t_1+1}{t}\right)^{t_1+1}\left(1-\frac{t_1+1}{t}\right)^{t_0} \\
&= \frac{1}{t^t}\left\{t_1^{t_1}(t_0+1)^{t_0+1} + (t_1+1)^{t_1+1}t_0^{t_0}\right\} .
\end{aligned}$$

Thus the prediction for the outcome $x_t = 1$ from its past x^{t-1} using the SNML strategy is given by

$$\begin{aligned}
P_{\text{SNML}}(x_t = 1 \mid x^{t-1}) &= \frac{P_{\hat{\boldsymbol{\theta}}(x^{t-1},1)}}{\sum_{x\in\mathscr{X}} P_{\hat{\boldsymbol{\theta}}(x^{t-1},x)}} \\
&= \frac{(t_1+1)^{t_1+1}t_0^{t_0}}{t_1^{t_1}(t_0+1)^{t_0+1} + (t_1+1)^{t_1+1}t_0^{t_0}} ,
\end{aligned} \tag{4.1}$$

and that for $x_t = 0$ is trivially given by

$$P_{\text{SNML}}(x_t = 0 \mid x^{t-1}) = 1 - P_{\text{SNML}}(x_t = 1 \mid x^{t-1}) .$$

In practice, instead of computing the SNML prediction, which could possibly result in overflow errors for large sample size, we can directly compute the SNML code length using the *log-sum-exp* trick. For our purpose, we also need to compute the encoded length of the event sequence x^n given event sequence y^n. Next we discuss how to conditionally encode one event sequence given the other using the past of both.

4.2.5.1 Causal Mechanism: Conditional SNML

To encode an event sequence x^n given an event sequence y^n, i.e. for $L(x^n \mid y^n)$, we have to compute $-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$. To predict the outcome x_t, we can use either x_i or y_i in every time step $i = 1, 2, \ldots, t-1$. Let $u = \sum_{i=1}^{t-1} x_i \oplus y_i$, with $\oplus$ being the Boolean XOR operator, be the number of time steps where the outcome of x_i and y_i differ. Thus we end up with 2^u different event sequence for predicting the outcome x_t. Among all possibilities, we choose the one that improves the prediction of x_t.

For exposition, we present a toy example in Eq. (4.2). Suppose we want to predict the outcome x_4 given its past $x^3 = 111$, and that of y^n, which is $y^3 = 010$. At every time step—except for the second— we have two choices. Overall we therefore can construct four different event sequence $z_1, \ldots, z_4$ that we can use to base our prediction on.

$$\begin{array}{lcccll}
x^3 & : & 1 & 1 & 1 & \\
y^3 & : & 0 & 1 & 0 & \\
z_1 & : & 0 & 1 & 0 & (y_3, 1, y_1) \\
z_2 & : & 0 & 1 & 1 & (y_3, 1, x_1) \\
z_2 & : & 1 & 1 & 1 & (x_3, 1, x_1) \\
z_4 & : & 1 & 1 & 0 & (x_3, 1, y_1)
\end{array} \tag{4.2}$$

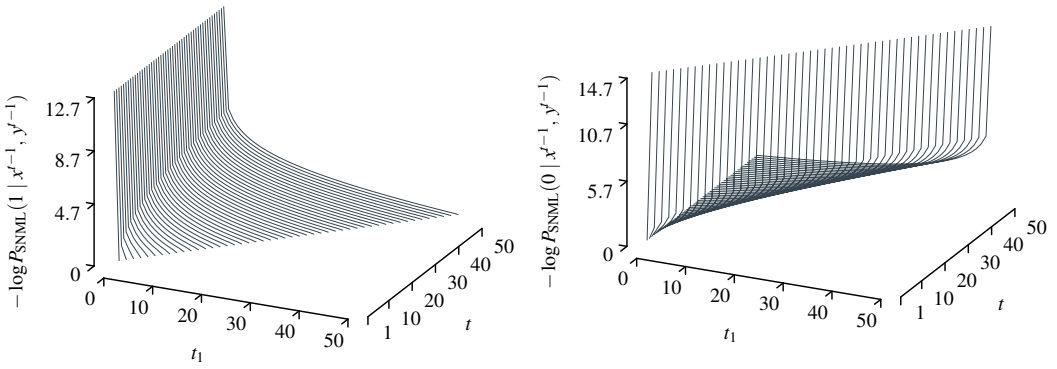


FIGURE 4.1: SNML predictions using the past of x^n, and that of y^n. For time steps $t = 1, 2, \ldots, 50$, and number of ones in the past $t_1 = 1, 2, \ldots, 50$ such that $t_1 < t$, we plot the code length using the SNML prediction strategy for (top) $x_t = 1$, and (bottom) $x_t = 0$.

By virtue of Eq. (4.1), we know that the prediction depends on the number of ones in the past t_1. Therefore we can use the number of ones present in newly constructed event sequence z_is to directly get the prediction. Let $t_x = \sum_{i=1}^{t-1} x_i$ be the number of ones in x^{t-1}, defining t_y analogous. To predict x_t given its past and that of y_n, we can use the number of ones in the range from $t_{\min} = \min(t_x, t_y)$ to $t_{\max} = \sum_{i=1}^{t-1} x_i \oplus y_i$.

In every iteration, we choose the number of ones $t_1 \in \{t_{\min}, t_{\min}+1, \ldots, t_{\max}\}$ that results in the minimal code length $-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$, using the prediction $P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$. This way, the optimisation of $-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$ also includes the index set for that of the $-\log P_{\text{SNML}}(x_t \mid x^{t-1})$. Thus we have

$$-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1}) \leq -\log P_{\text{SNML}}(x_t \mid x^{t-1}) \,.$$

The equality holds when y^{t-1} does not help in the prediction of x_t. As a result, both causal indicators are positive, i.e. $\Delta_{x^n \to y^n} \geq 0$, and $\Delta_{y^n \to x^n} \geq 0$.

At first glance, it is not evident whether the prediction is monotone with respect to the number of ones in the past t_1. Moreover, derivative analysis or an inductive proof appears to be non-trivial. Therefore we numerically compute the code length of the outcome x_t using the prediction $P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$ for various number of ones $t_1 = 1, 2, \ldots, t-1$ for a fixed t. Further we repeat the same process for $t = 1, 2, \ldots, 50$. In Fig. 4.1, we show the results in a 3D plot. We observe that the code length for the outcome $x_t = 1$ is monotonically decreasing with respect to the number of ones in the past t_1 for a fixed t. The code length for the outcome $x_t = 0$, however, is monotonically increasing relative to t_1 for a fixed t.

This numerical analysis suggests that $t_{\max}$ maximises the prediction of the outcome $x_t = 1$ given its past x^{t-1}, and that of y^n. On the contrary, $t_{\min}$ maximises the prediction of the outcome $x_t = 0$. Hence, the prediction for the outcome $x_t = 1$ from its past x^{t-1}, and that of y^n using SNML strategy is given by

$$P_{\text{SNML}}(x_t = 1 \mid x^{t-1}, y^{t-1}) = \frac{Z}{t_{\max}^{t_{\max}}(t_0+1)^{t_0+1} + Z} \,, \tag{4.3}$$

where $Z = (t_{\max}+1)^{t_{\max}+1} t_0^{t_0}$, and $t_0 = t - 1 - t_{\max}$. Likewise, the prediction for the outcome

$x_t = 0$ from its past x^{t-1}, and that of y^n using SNML strategy is given by

$$P_{\text{SNML}}(x_t = 0 \mid x^{t-1}, y^{t-1}) = \frac{K}{K + t_{\min}^{t_{\min}}(t_0+1)^{t_0+1}}\,, \tag{4.4}$$

where $K = t_{\min}^{t_{\min}}(t_0+1)^{t_0+1}$, and $t_0 = t - 1 - t_{\min}$.

From here onwards, we refer to the proposed framework as CUTE, for **ca**u**s**al inference on even**t** sequ**e**nces using SNML. All logarithms are to base 2, and by convention we use $0 \log 0 = 0$.

4.2.6 Computational Complexity

To compute $L(x^n)$, we have to compute $-\log P_{\text{SNML}}(x_t \mid x^{t-1})$ for $t = 1, 2, \ldots, n$. In every iteration, we can keep track of the count of the number of ones t_1 we have seen so far. Given t and t_1, we can compute $-\log P_{\text{SNML}}(x_t \mid x^{t-1})$ in constant time, $O(1)$, using the closed form expression in Eq. (4.1). Therefore we can compute $L(x^n)$ in $O(n)$ time.

To compute $L(x^n \mid y^n)$, we have to compute $-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$ for $t = 1, 2, \ldots, n$. In every iteration, we can keep track of the count of number of ones t_x, t_y, and $t_{\max}$. Given t_x, t_y, and $t_{\max}$, we can compute $-\log P_{\text{SNML}}(x_t \mid x^{t-1}, y^{t-1})$ in constant time, $O(1)$, using the closed form expression in Eq. (4.3), and Eq. (4.4). Hence we can compute $L(x^n \mid y^n)$ in $O(n)$ time. This implies we can compute - $\Delta_{x^n \to y^n}$ in $O(n)$ time.

Altogether the worst case computational complexity of the framework is $O(n)$.

4.3 Related Work

Causal inference techniques on time series are, for the most part, based on Granger causality (Granger, 1969). The key idea is that a time series x^n does not Granger cause a time series y^n if the past of x^n does not help in predicting y^n given the past of y^n. Typically predictability in measured in terms of the variance of the error in regression. This also corresponds to a significance test assuming a multivariate time series model (Chu and Glymour, 2008; Quinn et al., 2011). There exists many variants of Granger causality depending on the assumed model, and the predictability measure.

Linear Granger causality, for instance, considers a vector autoregressive (VAR) model. A VAR model describes the current outcome as a linear function of its past values, and an additive error term. Non-linear Granger causality is an extension of Granger causality to non-linear systems (Chen et al., 2004). The key idea there is to apply linear regression for each local neighbourhood and average the resulting statistical quantity over the entire attractor (a set of numerical values toward which a system tends to evolve). Rissanen and Wax (1987) proposed a compression-based framework for measuring mutual and causal dependence on the foundations of Granger causality, with an instantiation for continuous real-valued data. Another variant of Granger causality is the transfer entropy (Schreiber, 2000), or TENT for short, which measures predictability in terms of Shannon entropy. Transfer entropy can, unlike others, detect both linear and non-linear causal influences.

There do exist techniques that take a different approach than Granger causality. Chu and Glymour (2008) propose conditional independence test on non-iid setting, and introduce the additive non-linear time series models (ANLTSM). It uses additive model regression as a general purpose non-linear conditional independence test. TS-LiNGAM (Hyvärinen et al.,

2008) considers the general case where causal influences can occur either instantaneously or with considerable time lags. It combines the non-Gaussian instantaneous model with autoregressive models for causal analysis. Peters et al. (2013) use restricted structural equation models, ANMs in particular, to find causal structures from time series. Huang and Kleinberg (2015) introduce a causal inference framework based on logical formulas where cause is a discrete variable, and effect is a continuous-valued variable.

Except for transfer entropy, all the frameworks above either work with continuous-valued or mixed time series data. The known variants of Granger causality for event sequences, or discrete-valued time series are either highly tailored for the problem at hand (Quinn et al., 2011), or just minor variations (Pötter and Blossfeld, 2001). Transfer entropy is close in spirit to CUTE, but unlike transfer entropy, we are robust to adversarial settings (e.g. misspecified model classes). Importantly CUTE runs in $O(n)$ time. TENT, on the other hand, takes $O(nk+2^k)$, where k is the lag value. As it is the most commonly applied Granger causality framework, we compare CUTE against TENT in the experiments.

4.4 Experiments

We implemented CUTE in Python and provide the source code, along with the used datasets, and synthetic dataset generator.[3] All experiments were executed single threaded on MacBook Pro with 2.5 GHz Intel Core i7 processor and 16 GB memory. We compare CUTE with transfer entropy which is both commonly regarded as the state of the art in Granger causality and straightforwardly applicable to event sequences.

4.4.1 Synthetic Data

To evaluate CUTE on data with the known ground truth, we use synthetic data. We sample a cause event sequence x^n, with $\mathscr{X} \in \{0,1\}$, of size $n = 1000$ in i.i.d. fashion from the Bernoulli distribution with a random parameter from the Uniform distribution $\mathscr{U}(0.1,0.5)$. To generate the effect event sequence y^n from the cause event sequence x^n, we use an ANM in binary arithmetic:

$$y_t = f(x^{t-1}, y^{t-1}) + \varepsilon_t \pmod 2$$

where $\varepsilon_t \in \{0,1\}$ is an i.i.d. noise from a Bernoulli distribution with a random parameter θ in $\mathscr{U}(0,0.3)$. The modulo operation ensures that noise has a flipping effect when $\varepsilon_t = 1$. In particular, we consider ANMs with following functional forms:

- **shift:** $y_t = x_{t-s} + \varepsilon_t \pmod 2$, where $s \sim \mathscr{U}(1,20)$
- **shift+invert:** $y_t = x_{t-s} \oplus 1 + \varepsilon_t \pmod 2$, where $s \sim \mathscr{U}(1,20)$
- **rule-based:**

$$y_t = \left\{ \begin{array}{ll} B(\theta = 0.5), & \text{if } x_{t-1} = y_{t-1} \\ x_{t-1}, & \text{otherwise} \end{array} \right\} + \varepsilon_t \pmod 2 .$$

Whereas the first functional form shifts the cause event sequence by s time steps, the second one additionally inverts the resulting shifted event sequence. To generate a shifted sequence, we generate cause event sequence of size $s+n$, and then slice the first s time steps of the

[3] `https://github.com/kailashbuki/pycute`

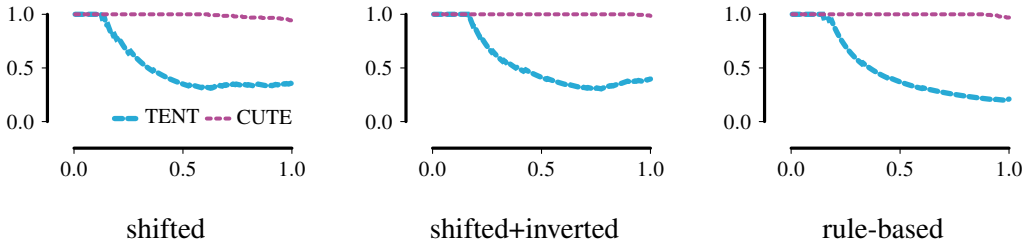


FIGURE 4.2: Accuracies at various decision rates on various functional forms.

cause after shifting the sequence. The last functional form uses a coin flip to generate outcomes of y^n if outcomes of x^n and y^n are the same at the previous time step, or else simply uses the outcome of x^n from the previous time step. In this case, we sample y_0 using a random coin flip.

In figure 4.2, we present accuracies of CUTE and TENT at increasing decision rates for 1000 cause-effect pairs for various functional forms aforementioned. The results show that CUTE is consistently more accurate than TENT in all cases, even at very high decision rates.

4.4.2 Real-World Data

Next we investigate CUTE on real-world data.

River Water Level

First we look into water level of rivers in Germany. We consider two rivers in particular: *Saar* and *Rhein*. For a river, we collect the raw water level recorded every 15 minutes from 25 June 2017 until 24 July 2017 from various water level recording stations.[4] This way we end up with 2880 data points from one station. The raw water level, however, is continuous real-valued, and hence we have to binarise the data. To this end, if the water level goes up from previous recording, we use 1. Otherwise we use 0. It is intuitively plausible to consider that the water level recording of the station upstream causes the water level recording of the station downstream.

For the *Saar* river, we collect the raw water level data from three stations, namely *Hanweiler*, *Sankt Arnual*, and *Fremersdorf*. Then we binarise the recordings from each station. In Figure 4.3, we show the map of the *Saar* river along with the recording stations. For instance, *Hanweiler* station is upstream compared to *Fremersdorf* station. Therefore the ground truth would be *Hanweiler* causes *Fremersdorf*. We run CUTE on all the pairs. In Figure 4.4, we present the results as a directed acyclic graph (DAG). The results clearly corroborate our intuition.

For the *Rhein* river, we collect the raw water level data from four stations, namely *Speyer*, *Mannheim*, *Worms*, and *Mainz*. Then we binarise the recordings from each station. In Figure 4.3, we show the map of the *Rhein* river along with the recording stations. After running CUTE on all the pairs, we end up with a DAG as shown in Figure 4.5. We see that CUTE identifies the correct direction in all but one case.

[4] `http://www.pegelonline.wsv.de/webservices/files/Wasserstand+Rohdaten/`

FIGURE 4.3: Map of the *Saar* and the *Rhein* river in Germany. The recording stations are marked with gray dots.

FIGURE 4.4: Results of CUTE on the *Saar* river. A green edge represents a correct causal direction.

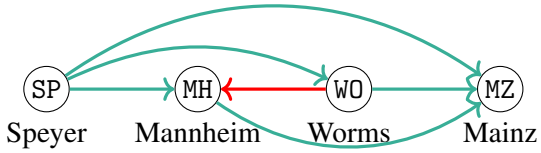


FIGURE 4.5: Results of CUTE on the *Rhein* river. A green edge represents a correct causal direction. A red edge indicates a wrong causal direction.

Temperature

The `temperature` dataset is the 48$^{\text{th}}$ pair in the Tübingen cause-effect benchmark pairs.[5] It contains indoor (x^n), and outdoor (y^n) temperature measurements recorded every 5 minutes. There are $n = 168$ measurements. We binarise the data like before. The ground truth of the pair is $y^n \to x^n$, which CUTE recovers correctly.

Overall, these results show that CUTE finds sensible causal directions from real data.

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

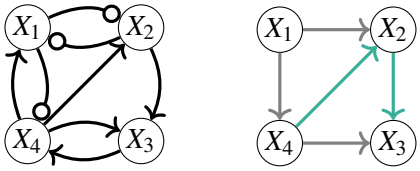


FIGURE 4.6: (left) Ground truth for the neural spike train data. A directed edge with a pointy head represents an excitatory influence, whereas a directed edge with a circular head represents an inhibitory influence. (right) Results of CUTE on the neural spike train data. A green edge represents a correct causal direction. A gray edge indicates a partially identified causal direction.

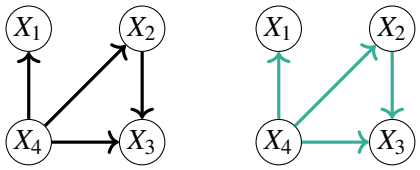


FIGURE 4.7: (left) Ground truth for the neural spike train data after removing the inhibitory influences, and cycles. (right) Results of CUTE on the modified neural spike train data. A green edge represents correct causal direction.

Neural Spike Train Recordings

Next we look at the neural spike train recordings data from an experiment carried out on a monkey (Quinn et al., 2011). As the original experimental data itself is not available for public use, we obtained the data simulator used by the authors to confirm their experimental data. The spike trains are generated using point process generalised linear models (GLM). The values of the parameters are selected to be within the range of parameters identified in point process GLM model fits to spike trains from electrode array recording data of primate primary motor cortex (Wu and Hatsopoulos, 2006). Typically there are two types of influences in neural spike trains, namely excitatory (neurons fire more) and inhibitory (neurons fire less).

In Fig. 4.6 (left), we show the ground truth of the spike train data. We note that CUTE is not designed to deal with feedback loops. The inferred DAG after testing pairwise causality using CUTE is presented on the right side of Fig. 4.6. We see that it correctly infers the direction of the two non-looped causal connections, and correctly recovers one of the two causal connections (as opposed to saying there is none) for X_1 and X_2, X_1 and X_4, and X_4 and X_3 each. When we remove the loops from the generating model by removing the inhibitory connections, we obtain the generating model depicted in Fig. 4.7. When we use CUTE to identify the causal direction of all dependent edges, we find that it recovers the ground truth.

4.5 Discussion

The experiments show that CUTE works well in practice. It reliably identifies true causal direction in a wide range of settings, is remarkably fast, and outperforms the state of the art by a wide margin, while qualitative case studies confirm that the results are sensible.

Although these results are promising, we see many possibilities for further improvement. We focused mainly on binary data in this work. But the extension to discrete data is also straightforward. We can compute P_{SNML} for a discrete time series by using the maximum likelihood estimator relative to a multinomial family. For the conditional compression, we can trivially extend the proposal in Section 4.2.5.1.

In this work, we do not take into account the *instantaneous* effects. In theory, however, we can include the instantaneous effects by using the current outcome y_t of the conditioned time series y^n in conditional compression cost $L(x^n \mid y^n)$. As such, we will then have $L(x^n \mid y^n) = \sum_{t=1}^{n} -\log P(x_t \mid x^{t-1}, y^t)$. Likewise, we can compute $L(x^n \mid y^n)$ using the causal mechanism proposed in Chapter 4.2.5.1.

Our method might fail in presence of a confounding time series z^n that causes both x^n and y^n. One of the avenues for future work would be to address that problem. That would also require reconsidering our assumptions, and computing compressed sizes of event sequences conditioned on the confounding event sequence. Last it would be interesting to explore the possibilities of extending CUTE on other data types, and using CUTE for causal discovery.

4.6 Conclusion

We proposed a causal inference framework for event sequences using information theory by building upon on the foundations of Granger causality. To encode the event sequences, we used minimax optimal codes relative to a parametric family of prediction strategies. We proposed CUTE, a linear time method for inferring the causal direction between two event sequences. Extensive evaluation on synthetic, and real-world data showed that CUTE discovers meaningful causal relations in binary-valued event sequences, and outperforms the state-of-the-art.

Software Artefacts

The Python implementation of causal inference methods used in this work has been released as a python package `pycute` in the PyPI repository.

Installation

The package requires Python $\geq$ 3.7. To install the package and all its dependencies, use `pip3`.

```
$ pip3 install pycute
```

Example Usage

For both CUTE and TENT, we report results in a tuple of the form $(\Delta_{x^n \to y^n}, \Delta_{y^n \to x^n})$.

```
>>> X, Y = [1] * 1000, [-1] * 1000
>>> from pycute import cute, tent
>>> cute.cute(X, Y)
>>> tent.tent(X, Y)
```

Bivariate Causal Inference on Multivariate IID Data 5

So far, we considered pairs of both i.i.d. and non-i.i.d. univariate discrete random variables. In practice, sometimes it is also of interest to know whether a group of variables together cause another group of variables. This, we consider next. In particular, we deal with the problem of inferring the causal direction between a pair of i.i.d. multivariate binary random variables.[1]

5.1 Introduction

Suppose that we have daily closing stock indices in various stock markets in Asia and Europe. For each of these markets we can record whether stock index went up from the previous closing or not (1 or 0). How can we tell if stock markets in one continent cause the other from this data? In a nutshell, we would like to infer the causal direction between a pair of multivariate binary random variables from a sample drawn from their joint distribution.

In recent years large strides have been made in the theory and practice of discovering causal structure from observational data (Peters et al., 2017b; Pearl, 2009; Hernán and Robins, 2019). Most methods, and especially those that defined for pairs of variables, however, can only consider continuous-valued or discrete numeric data (Shimizu et al., 2006; Peters et al., 2010; Janzing et al., 2012; Peters et al., 2014; Sgouritsa et al., 2015; Bloebaum et al., 2018) and are hence not applicable on multivariate binary data.

In this work, we instantiate the algorithmic independence of conditionals (AIC) through the crude version of the Minimum Description Length principle, and propose ORIGO,[2] which is a method for causal inference on multivariate binary data. In particular, we model the causal mechanism by a set of decision trees allowing for impure leaves in a decision tree. As such, we assume that effect is a non-deterministic boolean function of its cause.

To identify the best set of decision trees for a set of variables, we use the MDL principle. To this end, we encode one variable at a time using a decision tree. Such a tree may split only on previously encoded variables. We use this mechanism to measure how much better we can compress the data of $\mathbf{Y}$ given the data of $\mathbf{X}$, simply by (dis)allowing the trees for $\mathbf{Y}$

[1] This work is published as Budhathoki and Vreeken (2016, 2018c).

[2] ORIGO is Latin for origin

to split on variables of $\mathbf{X}$, and vice versa. We identify the most likely causal direction as the one with the most succinct description. Extensive experiments on synthetic, benchmark, and real-world data show that ORIGO is robust to noise, dimensionality, and skew between dimensionality of variables.

5.2 Crude MDL-based Approximation of AIC

Consider a pair of dependent multivariate binary random variables $\mathbf{X}$ and $\mathbf{Y}$. Suppose that we have a sample drawn from their joint distribution $P(\mathbf{X}, \mathbf{Y})$. From this sample, we would like to infer whether $\mathbf{X}$ causes $\mathbf{Y}$, or $\mathbf{Y}$ causes $\mathbf{X}$. To this end, we model the causal mechanism by a set of decision trees, and build upon the algorithmic independence of conditionals (AIC). This time, however, we consider crude MDL (Rissanen, 1978) codes to instantiate the AIC as it is intractable to compute stochastic complexity for rather complex model classes, such as a set of decision trees.

5.2.1 Crude MDL

Although the refined version of MDL possesses minimax optimality properties and leaves no room for arbitrariness, we can compute it in practice only for limited model classes, such as the exponential family. For rather complicated model classes, we have to resort to crude MDL. Despite arbitrariness, carefully designed two-part MDL codes are known to converge to the true distribution—if it exists—fast even on small sample sizes (Grünwald, 2007). The crude version of MDL, also known as two-part MDL, can be roughly described as follows (Grünwald, 2007).

Minimum Description Length Principle. *Given a set of models $\mathscr{M}$ and data D, the best model $M \in \mathscr{M}$ is the one that minimises*

$$L(D,M) = L(M) + L(D \mid M) \,,$$

where, in bits,

- $L(M)$ *is the length of the description of M, and*
- $L(D \mid M)$ *is the length of the description of D when encoded with M.*

Intuitively $L(M)$ represents the compressible part of the data, and $L(D \mid M)$ represents the noise in the data. In general, a model is a probability measure, and the set of models is a parametric collection of such models. Note that MDL requires the compression to be lossless in order to allow for fair comparison between different models $M \in \mathscr{M}$.

5.2.2 Approximating AIC by Crude MDL

The algorithmic independence of conditionals is based on the premise that we have access to the *true* distribution. In practice, we of course do not know this distribution, we only have observed data. MDL eliminates the need for assuming a distribution, as it instead identifies the model from the class that best describes the data. The total encoded size, which takes into account both how well the model fits the data as well as the complexity of the model, therefore functions as a practical instantiation of $K(P(\bullet))$.

To perform causal inference by MDL, we will need a model class $\mathscr{M}$ of causal models. Let $M_{\mathbf{X}\to\mathbf{Y}} \in \mathscr{M}$ be the causal model from the direction $\mathbf{X}$ to $\mathbf{Y}$. The causal model $M_{\mathbf{X}\to\mathbf{Y}}$ consists of model $M_{\mathbf{X}}$ for $\mathbf{X}$ and $M_{\mathbf{Y}|\mathbf{X}}$ for $\mathbf{Y}$ given $\mathbf{X}$. We define $M_{\mathbf{Y}\to\mathbf{X}}$ analogously. The total description length for the data over $\mathbf{X}$ and $\mathbf{Y}$ in the direction $\mathbf{X}$ to $\mathbf{Y}$ is given by

$$L_{\mathbf{X}\to\mathbf{Y}} = \underbrace{L(\mathbf{X}, M_{\mathbf{X}})}_{L(M_X)+L(\mathbf{X}|M_{\mathbf{X}})} + \underbrace{L(\mathbf{Y}, M_{\mathbf{Y}|\mathbf{X}} \mid \mathbf{X})}_{L(M_{\mathbf{Y}|\mathbf{X}})+L(\mathbf{Y}|M_{\mathbf{Y}|\mathbf{X}},\mathbf{X})} ,$$

where the first term is the total description length of $\mathbf{X}$ and $M_{\mathbf{X}}$, and the second the total description length of $\mathbf{Y}$ and $M_{\mathbf{Y}|\mathbf{X}}$ given the data of $\mathbf{X}$. We define $M_{\mathbf{Y}\to\mathbf{X}}$ analogously. Using the above indicators in the algorithmic independence of conditionals, we arrive at the following causal inference rules:

- If $L_{\mathbf{X}\to\mathbf{Y}} < L_{\mathbf{Y}\to\mathbf{X}}$, we infer $\mathbf{X} \to \mathbf{Y}$.
- If $L_{\mathbf{X}\to\mathbf{Y}} > L_{\mathbf{Y}\to\mathbf{X}}$, we infer $\mathbf{Y} \to \mathbf{X}$.
- Undecided otherwise.

That is, if total description length from $\mathbf{X}$ towards $\mathbf{Y}$ is simpler than vice versa, we infer $\mathbf{X}$ is likely the cause of $\mathbf{Y}$ under the causal mechanism represented by the model class. If it is the other way around, we infer $\mathbf{Y}$ is likely the cause of $\mathbf{X}$. If the total description length is the same in both directions, we are undecided. In practice, we can always introduce a threshold τ and treat differences between the two indicators smaller than τ as undecided. To use these indicators in practice, we have to define what causal model class $\mathscr{M}$ we use, how to describe a model $M \in \mathscr{M}$ in bits, how to encode a dataset D given a model M, and how to efficiently infer the optimal $M^* \in \mathscr{M}$. This we discuss in the next section.

5.2.3 Decision Trees as Causal Mechanism

A decision tree for a non-deterministic boolean function $f : \{0,1\}^m \to \{0,1\}$ is a binary tree whose internal nodes are labelled by the variables $X_1, \ldots, X_m$, and two outgoing edges from a node are labelled by 0 and 1. The leaves of the tree have probabilities of binary class labels. We assume that effect is a non-deterministic function of its cause, that can be represented by a set of decision trees.

In Figure 5.1, we give a toy example to show the valid models. For $M_{\mathbf{X}}$ and $M_{\mathbf{Y}}$, we only allow dependencies between variables in $\mathbf{X}$, and between variables in $\mathbf{Y}$ respectively, but not in between. In $M_{\mathbf{Y}|\mathbf{X}}$, we only allow variables in $\mathbf{Y}$ to acyclically depend on each other, as well as on variables in $\mathbf{X}$. Therefore, for the causal model $M_{\mathbf{X}\to\mathbf{Y}}$, we allow variables in $\mathbf{X}$ to depend on each other, and variables in $\mathbf{Y}$ to depend on either $\mathbf{X}$ or $\mathbf{Y}$. The reverse model $M_{\mathbf{Y}\to\mathbf{X}}$ is constructed analogously.

5.2.4 MDL-based Decision Trees

PACK (Tatti and Vreeken, 2008) is an MDL-based algorithm for discovering itemsets that compress the binary data efficiently. To do so, it discovers a set of decision trees that together encode the data most succinctly. While we do not care about these itemsets, it is the decision tree model PACK infers that is of interest to us.

For example, consider a toy binary dataset with three variables X_1, X_2, and X_3. PACK aims at discovering the set of decision trees such that we can encode the data in as few bits

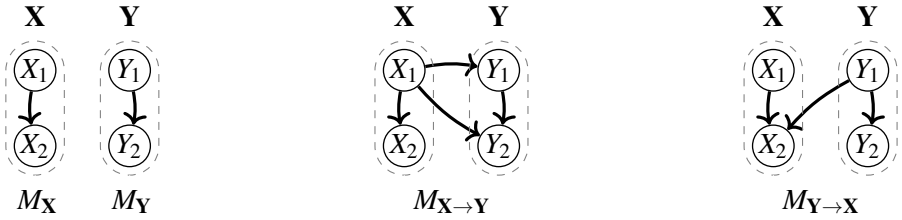


FIGURE 5.1: A toy example of valid causal models. A directed edge from a node u to a node v indicates that u depends on v.

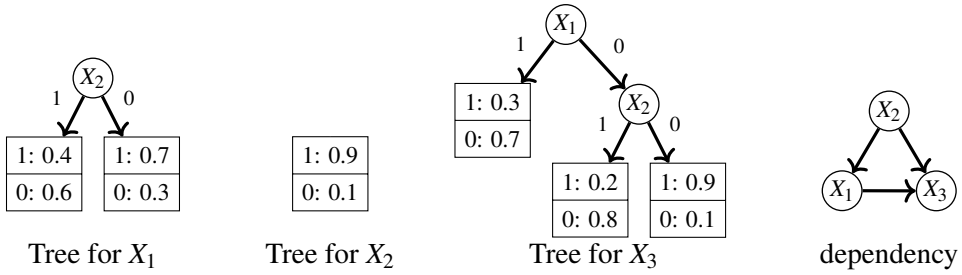


FIGURE 5.2: The three decision trees from the left are generated by PACK for a toy binary dataset with three variables X_1, X_2 and X_3. In the rightmost figure, we show dependencies between the variables based on their decision trees in a directed acyclic graph.

as possible. We give an example of the decision trees PACK could discover in Figure 5.2 (first three from the left). As the figure shows, X_1 depends on X_2, and X_3 depends on both X_1 and X_2. Those dependencies are shown in a directed acyclic graph in Figure 5.2 (rightmost).

For self-containment, we briefly explain how to compress binary data using MDL-based decision trees from Tatti and Vreeken (2008). Suppose that we have a sample of m binary random variables $\mathbf{X} = \{X_1, \ldots, X_m\}$, and the set of decision trees $M_{\mathbf{X}} = \{T_1, \ldots, T_m\}$ corresponding to each variable X_i. We use optimal prefix codes to encode each variable X_i by its corresponding decision tree T_i. To this end, using each leaf $l \in \text{leaves}(T_i)$ of the tree T_i, we encode a different part of X_i. As such, the total cost of encoding X_i using T_i is given by

$$L(X_i \mid T_i) = -\sum_{l \in \text{leaves}(T_i)} \sum_{v \in \{0,1\}} n_{(vl)} \log \hat{P}(X_i = v \mid l) \ ,$$

where $n_{(vl)}$ is the number of observations in leaf l that assume a value v, and $\hat{P}(X_i = v \mid l)$ is the empirical probability of the event $X_i = v$ given that leaf l is chosen.

To decode a variable, we have to transmit its corresponding decision tree as well. First we transmit leaves of a decision tree. Using refined MDL, we compute the encoded size of a leaf l as

$$L(l) = \log \sum_{j=0}^{n_{(l)}} \binom{n_{(l)}}{j} \left(\frac{j}{n_{(l)}}\right)^{n_{(l)}} \left(\frac{n_{(l)} - j}{n_{(l)}}\right)^{n_{(l)}-j} \ ,$$

where $n_{(l)}$ is the number of observations for which leaf l is used. For a parametric family of multinomial distributions, we can compute the expression above in linear time (Kontkanen and Myllymäki, 2007).

Then we encode the number of nodes in the decision tree T_i. In doing so, we use one bit to indicate whether a node is a leaf or an intermediate node. If a node is an intermediate

node, we use an extra $\log m$ bits to identify the split variable. Let $\text{inter}(T_i)$ be the set of all intermediate nodes of the decision tree T_i. Then the total number of bits needed to encode a decision tree T_i is given by

$$L(T_i) = \sum_{\text{node} \in \text{inter}(T_i)} \big(1 + \log m\big) + \sum_{l \in \text{leaves}(T_i)} \big(1 + L(l)\big) \ .$$

Thus the total encoded size of the decision tree T_i and that of X_i using T_i is given by

$$L(X_i, T_i) = L(T_i) + L(X_i \mid T_i) \ .$$

Altogether the total encoded size of all the decision trees $\mathscr{T}$ and all the variables using their corresponding decision trees is given by

$$L(\mathbf{X}, M_{\mathbf{X}}) = \sum_{T_i \in M_{\mathbf{X}}} L(X_i, T_i) \ .$$

To infer the best set of decision trees from the sample, PACK uses a greedy heuristic, which can be roughly described as follows. We start with a set of trivial decision trees without a split, one for each variable. For each variable X_i, we keep looking for the best variable X_j to split on that we have not split on before, as long as the dependency graph remains acyclic, and the total encoded size decreases. For more details on PACK, we refer the interested reader to Tatti and Vreeken (2008).

5.2.5 Instantiating the MDL score with PACK

To compute $L(X, M_{\mathbf{X}})$, we can simply compress $\mathbf{X}$ using PACK. Computing $L(Y, M_{\mathbf{Y}|\mathbf{X}} \mid \mathbf{X})$, however, is not straightforward, as PACK does not support conditional compression off-the-shelf. Clearly, it does not suffice to simply compress $\mathbf{X}$ and $\mathbf{Y}$ together as this gives us $L(\mathbf{XY}, M_{\mathbf{XY}})$ which may use any acyclic dependency between $\mathbf{X}$ and $\mathbf{Y}$ and vice versa. When computing $L_{\mathbf{X} \to \mathbf{Y}}$, for instance, we do not want the variables of $\mathbf{X}$ to depend on the variables of $\mathbf{Y}$. Therefore, we modify PACK such that a variable of $\mathbf{X}$ is only allowed to split on other variables of $\mathbf{X}$, and a variable of $\mathbf{Y}$ is allowed to split on both the variables of $\mathbf{X}$ and the other variables of $\mathbf{Y}$.

From here onwards, we refer to the PACK-based instantiation of the causal score as ORIGO, which means *origin* in latin. Although our focus is primarily on binary data, we can infer causal direction from categorical data as well. To this end, we can binarize the categorical data creating a binary feature per value. The implementation of PACK already provides this feature off-the-shelf.

5.2.6 Computational Complexity

Next we analyse the computational complexity of ORIGO. To compute $L_{\mathbf{X} \to \mathbf{Y}}$, we have to run PACK only once. PACK uses the ID3 algorithm to construct binary decision trees, therewith the computational complexity of PACK is $O(2^k n)$, where n is the number of observations in the sample, and k is the total number of univariate variables in $\mathbf{X}$ and $\mathbf{Y}$. To infer the causal direction, we have to compute both $L_{\mathbf{X} \to \mathbf{Y}}$, and $L_{\mathbf{Y} \to \mathbf{X}}$. Therefore, in the worst case, the computational complexity of ORIGO is $O(2^k n)$. In practice, ORIGO is fast, and completes within seconds.

5.3 Related Work

Most causal inference methods for a pair of variables are for univariate cause-effect pairs. Those methods typically exploit sophisticated properties of the joint distribution. One of the widely used frameworks are the Additive Noise Models (ANMs) (Shimizu et al., 2006; Hoyer et al., 2009). The ANMs assume that effect is a deterministic function of its cause and an additive noise that is independent of the cause. Although there exists variants of ANMs (Zhang and Hyvärinen, 2009; Peters et al., 2010), they are only for applicable on univariate cause-effect pairs.

Causal inference methods based on the algorithmic independence of conditionals (Peters et al., 2017b; Janzing and Schölkopf, 2010; Lemeire and Dirkx, 2006) require computable alternatives to Kolmogorov complexity. The information-geometric approach (Janzing et al., 2012) defines independence in terms of the orthogonality in information space. Sgouritsa et al. (2015) define independence in terms of the accuracy of the estimation of conditional distribution using corresponding marginal distribution. Liu and Chan (2016) define independence in terms of the distance correlation between empirical distributions, and propose DC. ERGO (Vreeken, 2015) is a causal inference framework based on relative conditional complexities, $K(Y \mid X)/K(Y)$ and $K(X \mid Y)/K(X)$, and infers the direction with the lowest relative complexity. Cumulative entropy is used to instantiate ERGO in practice for univariate and multivariate continuous real-valued data.

Causal inference on a pair of multivariate variables has received relatively much less attention. The linear trace method (Janzing et al., 2010; Zscheischler et al., 2011) infers linear causal relations of the form $Y = AX$, where A is the structure matrix that maps the cause to the effect, using the linear trace condition which operates on A, and the covariance matrix of X, Σ_X. The kernelized trace method (Chen et al., 2013) extends the trace-based method to non-linear causal relations.

Overall, only DC and ERGO are directly applicable to multivariate binary data.

5.4 Experiments

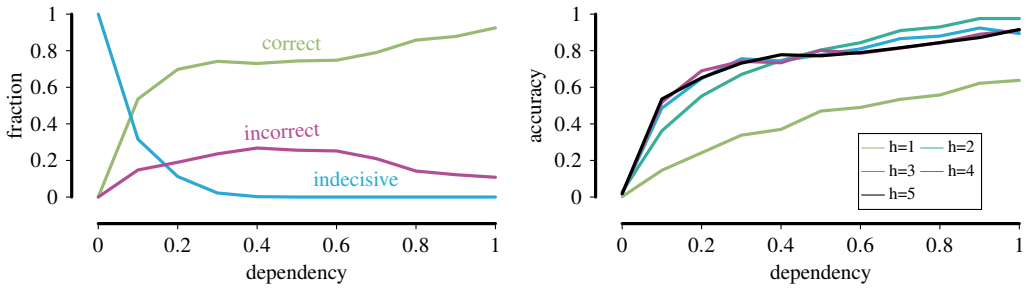
We implemented ORIGO in Python and provide the source code along with the used datasets, and synthetic dataset generator.[3] All experiments were executed single-threaded on MacBook Pro with 2.5 GHz Intel Core i7 processor and 16 GB memory running Mac OS X. We compare ORIGO against the ERGO score (Vreeken, 2015) instantiated with PACK, and DC (Liu and Chan, 2016).

5.4.1 Synthetic Data

To evaluate ORIGO on the data with known ground truth, we consider synthetic data. In particular, we generate binary data $\mathbf{X}$ and $\mathbf{Y}$ such that variables in $\mathbf{Y}$ probabilistically depend on the variables of $\mathbf{X}$, termed here onwards as *dependency*. Throughout the experiments on synthetic data, we generate $\mathbf{X}$ of size 5000-by-m, and $\mathbf{Y}$ of size 5000-by-p.

To this end, we generate data on a per variable basis. First, we assume the ordering of variables – the ordering of variables in $\mathbf{X}$ followed by the ordering of variables in $\mathbf{Y}$. Then, for each variable, we generate a binary decision tree. In doing so, we only consider the

[3] `https://github.com/kailashbuki/origo`



dependency vs various metrics dependency vs accuracy

FIGURE 5.3: For synthetic datasets with $m = p = 3$, we report (left) fraction of correct, incorrect and indecisive decisions at various dependencies, (right) the accuracy over all pairs at various dependencies for trees at various maximum heights.

variables preceding it in the ordering as candidate nodes for its decision tree. Then, each row is generated by following the ordering of variables, and using their corresponding decision trees. Further, we use the *split probability* to control the depth/size of the tree. We randomly choose weighted probabilities for the presence/absence of leaf variables.

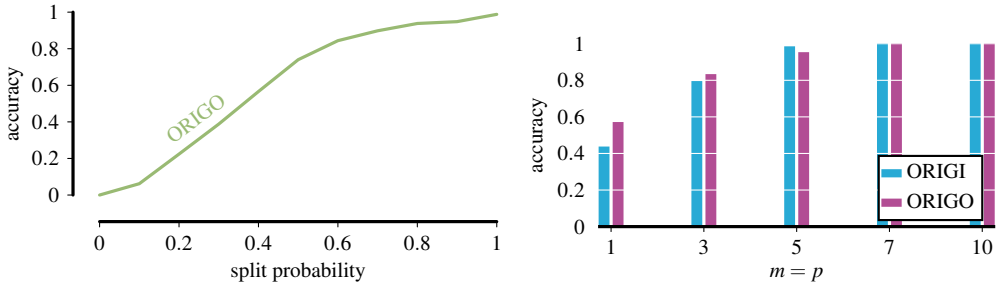
With the above scheme, with high probability, we generate data with a strong dependency in one direction. In general, we expect this direction to be the true causal direction, i.e. $\mathbf{X} \to \mathbf{Y}$. Although unlikely, it is possible, that the model in the reverse direction is superior. Moreover, unless we set the split probability to 1.0, however, it is possible that by chance we generate pairs without dependencies, and hence without a true causal direction. Unless stated otherwise we choose not to control for either case, by which at worst we underestimate the performance of ORIGO.

All reported values are averaged over 500 samples unless stated otherwise.

Performance: First we examine the effect of dependency on various metrics – the percentage of correct inferences (*accuracy* over all pairs), the percentage of indecisive inferences, and the percentage of incorrect inferences. We start with $m = p = 3$. We fix the split probability to 1.0, and generate trees with the maximum possible height, i.e. $m+p-1 = 5$. In Figure 5.3 (left), we give the plot showing various metrics at various dependencies for the generated pairs. We see that with the increase in dependency, indecisiveness quickly drops to zero, while accuracy increases sharply towards 90%. Note that at zero dependency, there are no causal edges, hence ORIGO is *correct* in being indecisive.

Next we study the effect of the maximum height h of the trees on the accuracy of ORIGO. We set $m = p = 3$, and the split probability to 1.0. In Figure 5.3 (right), we observe that the accuracy gets higher as h increases. This is due to the increase in the number of causal edges with the increase in the maximum height of the tree. Although the increase in accuracy is quite large when we move from $h = 1$ to 2, it is almost negligible when we move from $h = 2$ onwards. This shows that ORIGO already infers the correct causal direction even when there are only few causal dependencies in the generating model.

Next we analyse the effect of split probability on the accuracy of ORIGO. To this end, we set $m = p = 3$, fix the dependency to 1.0, and generate trees with the maximum possible height. In Figure 5.4 (left), we observe that the accuracy of ORIGO increases with the increase in the split probability. This is due to the fact that the depth of the tree increases with



split probability vs accuracy ORIGI vs ORIGO

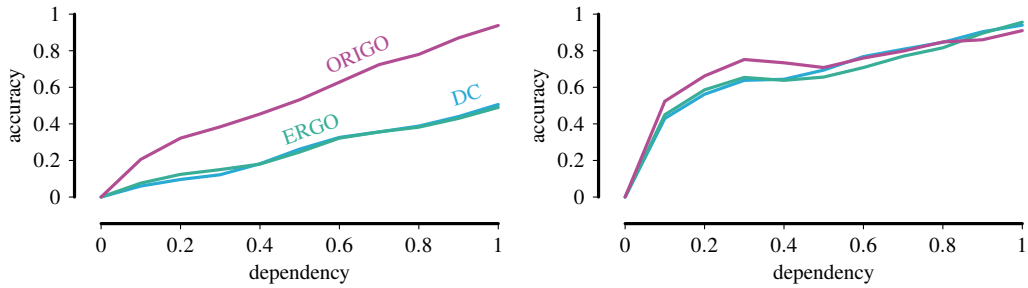
FIGURE 5.4: For synthetic datasets, we show (left) the accuracy over all pairs at various split probabilities for ORIGO with $m = p = 3$, and (right) compare the accuracy over all pairs against bagging in symmetric case with $m = p$.

the increase in the split probability. Consequently, there are more causal edges, therewith the more accurate ORIGO is.

Next, we examine whether considering a rather large space of data instead of single sample improves the result. To this end, we perform bootstrap aggregating, also called *bagging*. Bagging is the process of sampling K new datasets D_i from a given dataset D uniformly and with replacement. We fix the dependency to 0.7, the probability of split to 1.0, the number of bagging samples to $K = 50$ and generate trees with maximum height of $h = 5$. We run ORIGO on each sampled cause-effect pair. Then we take the majority vote to decide the causal direction. In Figure 5.4 (right), we compare the accuracy of ORIGO against bagging (ORIGI) for symmetric cause effect pairs. We see that bagging does not really improve the result. This is not unexpected as bagging is mainly a way to overcome overfitting, which by MDL we are naturally protected against (Grünwald, 2007). These results confirm this conviction.

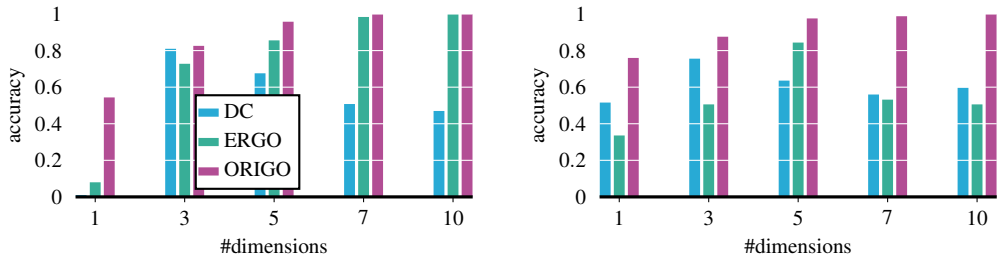
Next we investigate the accuracy of ORIGO on cause-effect pairs with asymmetric dimensions. For that, we fix the split probability to 1.0, and generate trees with the maximum possible height. At every level of dependency, we generate 500 cause-effect pairs, 250 of which with $m = 1, p = 3$ and remaining 250 with $m = 3, p = 1$. In particular, we consider those pairs for correctness where there is at least one causal edge from $\mathbf{X}$ to $\mathbf{Y}$. In Figure 5.5 (left), we give the plot comparing the accuracy of ORIGO against ERGO and DC. We see that ORIGO performs much better than the other methods. In particular, the difference in accuracy gets larger as the dependency increases. We also note that the performance of DC has a striking resemblance to flipping a fair coin.

Next we consider the symmetric case where $m = p = 3$. We fix the split probability to 1.0, and generate trees with the maximum possible height. As in the asymmetric case, we consider those pairs for correctness where there is at least one causal edge from $\mathbf{X}$ to $\mathbf{Y}$. In Figure 5.5 (right), we show the plot comparing the accuracy of ORIGO against ERGO, and DC. We see that both ORIGO performs as good or better than other methods. We note that for the pairs without dependency, DC infers a causal relationship in over 50% of the cases.



asym. dim.: 1 vs 3 $m = p = 3$

FIGURE 5.5: For synthetic datasets, we compare (left) the accuracy over all pairs in asymmetric case (1 vs. 3), and (right) the accuracy over all pairs at various dependencies in symmetric case ($m = p = 3$)



symmetric case, $m = p$ asymmetric case

FIGURE 5.6: For synthetic datasets, we report the accuracy over all pairs (left) in symmetric case with $m = p$, and (right) in asymmetric case (5 vs. varying dimensions).

Dimensionality

Next we study the robustness against dimensionality. First we consider cause-effect pairs with symmetric number of dimensions, i.e. $m = p$ and vary it between 1 and 10. We fix the dependency to 0.7, the split probability to 1.0, and the maximum height of trees to 5. In particular, we compare ORIGO against ERGO and DC. In Figure 5.6 (left), we see that ORIGO is highly accurate in every setting. With the exception of the univariate case, ERGO also performs well when both $\mathbf{X}$ and $\mathbf{Y}$ have the same cardinality.

In practice, however, we also encounter cause-effect pairs with asymmetric number of dimensions. To evaluate performance in this setting, we set respectively m and p to 5 and vary the other between 1 to 10 – and generate 100 data pairs per setting. In Figure 5.6 (right), we see that ORIGO outperforms ERGO by a huge margin the stronger the imbalance between the cardinalities of $\mathbf{X}$ and $\mathbf{Y}$. This is due to the inherent bias of ERGO favouring the causal direction from the side with higher complexity towards the simple one. In addition, we see that ORIGO outperforms DC in every setting.

5.4.2 Real-World Data

Next, we evaluate ORIGO on real-world data.

Table 5.1: Results on the benchmark multivariate cause-effect pairs. A tick (✓) indicates a correct decision, a cross (×) indicates a wrong decision, and a double-headed arrow (↔) indicates an indecision.

Dataset	n	m	p	Truth	ORIGO	ERGO	DC
weather forecast	10,226	4	4	$\mathbf{Y} \to \mathbf{X}$	↔	✓	↔
ozone	989	1	3	$\mathbf{Y} \to \mathbf{X}$	✓	✓	×
auto-mpg	392	3	2	$\mathbf{X} \to \mathbf{Y}$	✓	✓	×
radiation	72	16	16	$\mathbf{Y} \to \mathbf{X}$	×	×	×
chemnitz	1,440	3	7	$\mathbf{X} \to \mathbf{Y}$	✓	×	✓
car	1,728	6	1	$\mathbf{X} \to \mathbf{Y}$	✓	✓	✓

Multivariate Benchmark Pairs

First we evaluate ORIGO on real-world data with multivariate pairs. For that we consider four cause-effect pairs with known ground truth taken from the Tübingen cause-effect benchmark pairs.[4] The `chemnitz` dataset is taken from Janzing et al. (2010), whereas the `car` dataset is from the UCI repository.[5] We use IPD (Nguyen et al., 2014) to discretize the data. We give the base statistics in Table 5.1. For each pairs, we report the sample size, the dimension of $\mathbf{X}$, the dimension of $\mathbf{Y}$, the ground truth. Furthermore, we report the results of ORIGO, ERGO, and DC. We observe that both ORIGO and ERGO infer correct direction from four pairs. Whereas ORIGO is incorrect in one pair and remains indecisive in the other, ERGO is incorrect in two pairs. DC, however, is mostly incorrect.

Acute inflammation

The `acute inflammation` dataset is taken from the UCI repository.[4] It consists of the diagnosis of two diseases of urinary system for 120 potential patients. There are 6 symptoms – temperature of the patient (X_1) , occurrence of nausea (X_2), lumber pain (X_3), urine pushing (X_4), micturition pains (X_5), burning of urethra, itch, swelling of urethra outlet (X_6). All the symptoms are binary but the temperature of the patient, which takes a real value between $35°C - 42°C$. The two diseases for diagnosis are inflammation of urinary bladder (Y_1) and nephritis of renal pelvis origin (Y_2). We discretise the temperature into two bins using IPD. This results in two binary attributes X_{11} and X_{12}. We then run ORIGO on the pair $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = \{X_{11}, X_{12}, X_3, X_4, X_5, X_6\}$ and $\mathbf{Y} = \{Y_1, Y_2\}$. We find that $\mathbf{Y} \to \mathbf{X}$. That is, ORIGO infers that the disease causes the symptoms, which is in agreement with our intuition.

5.5 Discussion

In this work, we showed how to instantiate the algorithmic independence of conditionals through the crude version of the Minimum Description Length principle for a pair of multivariate binary random variables. The experiments show that ORIGO works well in

[4] `https://webdav.tuebingen.mpg.de/cause-effect/`
[5] `https://archive.ics.uci.edu/ml/`

practice. ORIGO reliably identifies true causal structure regardless of cardinality, skew, with high statistical power, even at low level of dependencies. Moreover, the qualitative case studies show that the results are sensible.

Although these results show the strength of our framework, and of ORIGO in particular, we see many possibilities to further improve. For instance, PACK does not work directly on categorical data. By binarizing the categorical data, it can introduce undue dependencies. This presents an inherent need for a lossless compressor that works directly on categorical data. For other data types, for instance, Marx and Vreeken (2019) have extended ORIGO to mixed-type data using crude MDL-based regression trees.

One avenue for future work would be to include the missing identifiability results for ORIGO, as it does not have any theoretical guarantees on the identifiability of the causal graph from the joint distribution of variables. Similar to other causal inference frameworks for a pair of variables, ORIGO is based on the causal sufficiency assumption. Extending ORIGO to include confounders is another avenue of future work.

5.6 Conclusion

We proposed a causal inference method for a pair of i.i.d. multivariate binary random variables. To this end, we instantiated the algorithmic independence of conditionals through MDL-based decision trees. We model the causal mechanism by a set of decision trees allowing for impure leaves in a decision tree. As we can compute refined MDL code only for limited few model classes, we computed the two-part MDL code of the data, and the set of decision trees. Extensive evaluation on synthetic, benchmark, and real-world data showed that ORIGO reliably infers the correct causal direction on a wide range of settings.

Software Artefacts

The Python implementation of MDL-based causal inference methods in this work has been released as a python package `origo` in the PyPI repository.

Installation

The package requires Python $\geq$ 3.7. To install the package and all its dependencies, use `pip3`.

```
$ pip3 install origo
```

Example Usage

We report results in a tuple of the form $(L_{\mathbf{X}\to\mathbf{Y}}, L_{\mathbf{Y}\to\mathbf{X}})$.

```
>>> X, Y = [[1],[1],...], [[-1],[-1],...]
>>> from origo import origo, ergo
>>> origo.origo(X, Y)
>>> ergo.ergo(X, Y)
```

Discovering Reliable Causal Rules 6

Just knowing that a group of variables cause a target variable does not always fully satisfy one's curiosity. It is often of particular interest, for instance, to know the conditions on the variables under which the effect is visible, such as the specific combinations of drugs that lead to severe side-effects. In this chapter, we study the problem of deriving rules or policies from observational data that, when enacted on a complex system, cause a desired outcome.[1]

6.1 Introduction

Consider the study of the effect of combinations of drugs. Certain drugs can amplify each others effect, and are therewith combinations of drugs can turn out to be much more effective, or even only effective, than when the drugs are taken individually. This effect is sometimes positive, for example in combination treatments against HIV and cancer, but sometimes it is also negative, as it can lead to severe up to possibly lethal side effects. For all but the smallest number of drugs, however, there are so many possible combinations that it quickly becomes practically impossible to test these combinations in a controlled manner. From the observational data, however, we can only establish that some drugs together cause the recovery. We need more than such a generic statement because some drug combinations are, for instance, more effective than the others. Some, on the other hand, can lead to severe side-effects. In such cases, it is of particular interest to us to identify the combination of drugs that are most effective.

In a nutshell, we would like to discover rules from a set of actionable variables that are most effective with regard to a target variable of interest. Existing methods for causal inference from observational data, however, can only extract partially directed causal graphs from data (Spirtes et al., 2000; Chickering, 2002; Pearl, 2009), or identify the most likely causal direction between pairs of variables (Shimizu et al., 2006; Hoyer et al., 2009). Though simple to state, this task of discovering effective rules is not only computationally hard; we also have have to cope with an intricate combination of two semantic problems—one statistical and one structural.

[1] This work was presented at the NeurIPS 2018 workshop on Causal Learning (Budhathoki et al., 2018), and the full research article is currently under submission.

The statistical problem is the well-known phenomenon of overfitting. This phenomenon results from the high variance of the naive empirical (or "plug-in") estimator of causal effect for rules with too small sample sizes for the instances either covered, or excluded by the rule. Combined with the maximization task over a usually very large rule language, this variance turns into a strong positive bias that dominates the search and causes essentially random results of either extremely specific or extremely general rules.

The second, structural problem is often referred to as Simpson's paradox. Even strong and confidently measured effects of a rule might not actually reflect true domain mechanisms, but can be mere artifacts of the effect of other variables. Notably, such confounding effects can not only attenuate or amplify the marginal effect of a rule on the target variable, in the most misleading cases they can even result in sign reversal, i.e. when interpreted naively, the data might indicate a negative effect even though in reality there is a positive effect (Pearl, 2009, Chap. 6).

In this work, we present a theoretically sound approach to discovering causal rules that remedies both of these problems.

1. To address the overfitting problem, we propose to measure and optimise the *reliable* effect of a rule. In contrast to the plug-in estimator, we propose a conservative empirical estimate of the population effect, that is not prone to overfitting. Additionally, and in contrast to other known rule optimisation criteria, it is also *consistent*, i.e., with increasing amounts of evidence (data), the measure converges to the actual population effect of a rule.

2. To address the structural problem, we propose to control for the effect of a given set of potential confounder variables. In particular, we identify the admissible data-generating processes under which it is possible to discover causal rules. While in practice the set of control variables will rarely be complete, i.e., not contain all potential confounders, this approach can rule out specific alternative explanations of findings as well as eliminate misleading observations caused by selected observables that are known to be strong confounders. In fact, this pragmatic approach is usually a necessity caused by the limited sample size.

3. We develop a practical algorithm for efficiently discovering the top-k strongest reliable effect rules. In particular, we show how the optimisation function can be cast into a branch-and-bound approach based on computationally efficient tight, optimistic estimator.

Moreover, our approach lends itself naturally to an iterative approach in which we can discover insights beyond the factors included in the control variables. We support our claims by experiments on real-world datasets as well as by reporting the required computation times on a large set of benchmark datasets.

6.2 Reliable Causal Rules

We consider a causal system of discrete (random) variables with a designated **target variable** Y and a number of covariates, which we differentiate into **actionable variables** $\mathbb{X}$ and **control variables** $\mathbb{Z}$. For example, Y might indicate recovery from a disease, $\mathbb{X}$ different medications that can be administered to a patient, and $\mathbb{Z}$ might contain patient properties

like sex and age. We assume that one can perform an intervention on the system that involves enforcing specific values $\mathbf{x}$ for some chosen set of variables $\mathbf{X} \subseteq \mathbb{X}$, after the values $\mathbf{z}$ for $\mathbb{Z}$ have been observed. Each such intervention yields different joint **post-intervention probabilities**, denoted by $\mathrm{P}(\bullet \mid do(\mathbf{x}), \mathbf{z})$ (Pearl, 2009, Chap. 3). We are interested in (stochastic) **policies** Q that enforce values $\mathbf{x}$ non-deterministically with probability $Q(\mathbf{x} \mid \mathbf{z})$. The post-intervention probabilities are then given as the mixture

$$\mathrm{P}(\bullet | do(Q), \mathbf{z}) = \sum_{\mathbf{x} \in \mathscr{X}} \mathrm{P}(\bullet | do(\mathbf{x}), \mathbf{z}) Q(\mathbf{x} \mid \mathbf{z}) \ ,$$

where $\mathscr{X}$ is the joint domain of $\mathbf{X}$. For example, a doctor might describe different medications to patients of different demographics and different doctors might have varying preferences for medications resulting in stochasticity when considering the policy of a hospital as whole. Note that in practice we usually have a choice as to which covariates to consider as actionable variables and which as control. Below we discuss in detail the considerations affecting this choice.

In this work, we are concerned with policies that are described by **rules** $\sigma(\mathbf{x})$ that evaluate to either true ($\top$) or false ($\bot$) for a given value $\mathbf{x}$. Specifically, we investigate the **rule language** $\mathscr{L}$ of conjunctions of **propositions** $\sigma \equiv \pi_1 \wedge \cdots \wedge \pi_l$) that can be formed from inequality and equality conditions on individual actionable variables X (i.e., $\pi \equiv X \leq v$ and so on). In general there are many values $\mathbf{x}$ that can satisfy a rule σ (e.g., $\sigma \equiv X \leq 3$ is satisfied by $X = 3, 2, \ldots$). Hence, such rules describe a multitude of compatible policies Q_σ, each of which enforces σ, i.e.,

$$\mathrm{P}(\sigma(\mathbf{X}) = \top \,|\, do(Q_\sigma), \mathbf{z}) = 1 \ .$$

When working with observational data that has been sampled according to some **observational probabilities**, this degree of freedom is eliminated by identifying the rule σ with the **observationally congruent policy** defined as $Q_\sigma(\mathbf{x} \mid \mathbf{z}) = \mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z})$. This specific choice permits truthful evaluation through the observational data.

Our goal is to identify rules σ that have a high **causal effect** on a specific **outcome** y for the target variable Y, which we define as the difference in the post-intervention probabilities of y under the policies described by σ and $\bar{\sigma}$, i.e.,

$$\overrightarrow{e}(\sigma) = \mathbb{E}\left[\mathrm{P}(y \mid do(\sigma), \mathbf{z}) - \mathrm{P}(y \mid do(\bar{\sigma}), \mathbf{z})\right] \ .$$

where $\bar{\sigma}$ denotes the logical negation of σ.

6.2.1 From Observational to Causal Effect

On observational data, we can approximate the causal effect by the (conditional) **observational effect** of σ on Y where we replace intervention by simple conditioning, i.e.,

$$e(\sigma) = \mathbb{E}\left[\mathrm{P}(y \mid \sigma, \mathbf{z}) - \mathrm{P}(y \mid \bar{\sigma}, \mathbf{z})\right] \ ,$$

where we use the shorthands σ and $\bar{\sigma}$ to denote the events $\sigma(\mathbf{X}) = \top$ and $\sigma(\mathbf{X}) = \bot$, respectively. However, unless special conditions hold, the observed conditional probabilities will not be the same as the post-intervention probabilities; hence $e(\sigma) \neq \overrightarrow{e}(\sigma)$. That is, in general, we only measure association rather than causation.

As mentioned in the introduction, a well-known reason for this discrepancy is the potential presence of **confounders**, i.e., variables that influence both, our desired actionable variable(s) and the target. More generally, to get accurate causal effect estimates, we have to eliminate the influence of all *spurious path* in the **causal graph**, i.e., the directed graph that describes the conditional independences of our random variables (with respect to all post-intervention distributions). Intuitively, we can achieve this by choosing our set of control variables large enough.

In more detail, when estimating the causal effect of X on Y, any undirected path connecting Y and X that has an incoming edge towards X is a **spurious path**. A node (variable) is a **collider** on a path if its in-degree is 2, e.g., Z is a collider on the path $X \rightarrow Z \leftarrow Y$. A spurious path is **blocked** by a set of nodes $\boldsymbol{Z}$, if the path contains a collider that is not in $\boldsymbol{Z}$, or a non-collider on the path is in $\boldsymbol{Z}$ (Pearl, 2009, Def. 1.2.3). A set of nodes $\boldsymbol{Z}$ satisfies the **back-door criterion** for a set of nodes $\mathbf{X}$ and a node Y if it blocks all spurious paths from any X in $\mathbf{X}$ to Y, and there is no direct path from any X in $\mathbf{X}$ to any Z in $\boldsymbol{Z}$ (Pearl, 2009, Def. 3.3.1). For $\mathbf{X}$ and Y, if a set $\boldsymbol{Z}$ satisfies the back-door criterion, then observation and post-intervention probabilities are equal within each $\mathbf{z}$ stratum of $\boldsymbol{Z}$ (Pearl, 2009, Thm. 3.3.2):

$$\mathrm{P}(y \mid do(\mathbf{x}), \mathbf{z}) = \mathrm{P}(y \mid \mathbf{x}, \mathbf{z}) \,. \tag{6.1}$$

Consequently, for $e(\sigma)$ to be a truthful estimate of $\overrightarrow{e}(\sigma)$ for all rules $\sigma \in \mathscr{L}$, we have to assure that our control variables $\mathbb{Z}$ satisfy the back-door criterion for *all* the actionable variables $\mathbb{X}$. This also implies that there are no other spurious paths via potentially unobserved variables $\boldsymbol{U}$. In the special case that $\mathbb{Z}$ is empty, Y must not cause any actionable variable X in $\mathbb{X}$. The following definition summarises all these requirements (see Fig. 6.1 for an illustration).

Definition 6.2.1 (Admissible Input to Causal Rule Discovery)**.** *The causal system* $(\mathbb{X}, Y, \mathbb{Z})$ *of actionable variables, target variable, and control variables is an admissible input to causal rule discovery if the underlying causal graph of the variables satisfy the following:*

(a) there are no outgoing edges from Y *to any* X *in* $\mathbb{X}$*,*

(b) no outgoing edges from any X *in* $\mathbb{X}$ *to any* Z *in* $\mathbb{Z}$*,*

(c) no edges between actionable variables $\mathbb{X}$*, and*

(d) no edges between any unobserved U *in* $\boldsymbol{U}$ *and* X *in* $\mathbb{X}$*.*

The lemma below shows that the control variables $\mathbb{Z}$ block all spurious paths between any subset of actionable variables $\mathbf{X}$ and Y.

Lemma 6.2.1. *Let* $(\mathbb{X}, Y, \mathbb{Z})$ *be an admissible input. Then the control variables* $\mathbb{Z}$ *block all spurious paths between any subset of actionable variables* $\mathbf{X}$ *and* Y*.*

Proof. To prove this lemma, we argue graphically. All spurious paths between $\mathbf{X}$ and Y can be either from Y to $\mathbf{X}$ directly, or via control variables $\mathbb{Z}$, or via other actionable variables $\mathbb{X} \setminus \mathbf{X}$, or via latent variables $\boldsymbol{U}$. Criterion (a) rules out trivial spurious paths from Y to $\mathbb{X}$ that cannot be blocked by any $\mathbb{Z}$. Criterion (b) ensures that any spurious path unblocked by one control variable is blocked by another. Criterion (c) ensures that there are no spurious

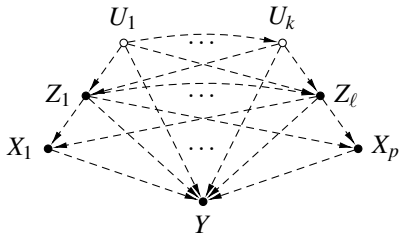


FIGURE 6.1: A skeleton causal graph of an admissible input to causal rule discovery (see Def. 6.2.1). A dashed edge from a node u to v indicates that u potentially affects v.

paths between any subset of actionable variables $\mathbf{X}$ in the rule σ and Y via other actionable variables $\mathbb{X} \setminus \mathbf{X}$. To see this, suppose that we have two actionable variables X_1 and X_2, and a rule $\sigma \equiv X_1 = 1$. If the causal graph contains the path $X_1 \leftarrow X_2 \rightarrow Y$, the observational effect is a biased estimator of the causal effect. Criterion (d) is really just a form of standard causal sufficiency (Scheines, 1997). In particular, as there are no edges between any latent variable U in $\boldsymbol{U}$ and any X in $\mathbb{X}$, by conditioning on $\mathbb{Z}$, we block any spurious path between any $\mathbf{X}$ and Y via $\boldsymbol{U}$. Thus the control variables $\mathbb{Z}$ block all spurious paths between any subset of actionable variables $\mathbf{X}$ and Y. □

The following theorem notes that, for admissible inputs, $e(\sigma)$ is equal to the causal effect of a rule or, more precisely, its observationally congruent policy (proof in appendix).

Theorem 1. *Let ($\mathbb{X}$, Y, $\mathbb{Z}$) be an admissible input. Then for any rule σ in rule language $\mathscr{L}$, we have $e(\sigma) = \overrightarrow{e}(\sigma)$.*

Proof. The post-intervention probability of y in a $\mathbf{z}$ stratum of $\mathbb{Z}$ under the policy described by σ is given by

$$\mathrm{P}(y \mid do(\sigma), \mathbf{z}) = \sum_{\mathbf{x} \in \mathscr{X}} \mathrm{P}(y \mid do(\mathbf{x}), \mathbf{z}) Q_\sigma(\mathbf{x} \mid \mathbf{z})$$

As σ describes the observationally congruent policy $Q_\sigma(\mathbf{x} \mid \mathbf{z}) = \mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z})$, we have

$$= \sum_{\mathbf{x} \in \mathscr{X}} \mathrm{P}(y \mid do(\mathbf{x}), \mathbf{z}) \mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z})$$

As $\mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z}) = 0$ for all $\mathbf{x}$ with $\sigma(\mathbf{x}) = \bot$, we can only take those $\mathbf{x}$ for which $\sigma(\mathbf{x}) = \top$ holds in the summation

$$= \sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \mathrm{P}(y \mid do(\mathbf{x}), \mathbf{z}) \mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z})$$

As $\mathbb{Z}$ blocks all spurious paths from Y to any $\mathbf{X} \subseteq \mathbb{X}$ due to Lemma 6.2.1, for any rule σ in $\mathscr{L}$, we can replace the post-intervention probability under intervention $do(\mathbf{x})$ by the observational conditional probability within each $\mathbf{z}$ stratum of $\mathbb{Z}$ using Eq. (6.1)

$$= \sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \mathrm{P}(y \mid \mathbf{x}, \mathbf{z}) \mathrm{P}(\mathbf{x} \mid \sigma, \mathbf{z})$$

$$= \sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \frac{\mathrm{P}(y \mid \mathbf{x}, \mathbf{z}) \mathrm{P}(\mathbf{x}, \sigma \mid \mathbf{z})}{\mathrm{P}(\sigma \mid \mathbf{z})}$$

For $\mathbf{x}$ with $\sigma(\mathbf{x}) = \top$, we have $\mathrm{P}(\mathbf{x}, \sigma \mid \mathbf{z}) = \mathrm{P}(\mathbf{x} \mid \mathbf{z})$, thus

$$= \sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \frac{\mathrm{P}(y \mid \mathbf{x}, \mathbf{z}) \mathrm{P}(\mathbf{x} \mid \mathbf{z})}{\mathrm{P}(\sigma \mid \mathbf{z})}$$

$$= \sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \frac{\mathrm{P}(y, \mathbf{x} \mid \mathbf{z})}{\mathrm{P}(\sigma \mid \mathbf{z})}$$

Since $\sum_{\substack{\mathbf{x} \in \mathscr{X} \\ \sigma(\mathbf{x}) = \top}} \mathrm{P}(y, \mathbf{x} \mid \mathbf{z}) = \mathrm{P}(y, \sigma \mid \mathbf{z})$, this results in

$$= \frac{\mathrm{P}(y, \sigma \mid \mathbf{z})}{\mathrm{P}(s \mid \mathbf{z})}$$

$$= \mathrm{P}(y \mid \sigma, \mathbf{z}) \, .$$

Using this result, we have the causal effect of σ on Y as

$$\begin{aligned} \overrightarrow{e}(\sigma) &= \mathbb{E}\left[\mathrm{P}(y \mid do(\sigma), \mathbf{z}) - \mathrm{P}(y \mid do(\bar{\sigma}), \mathbf{z})\right] \\ &= \mathbb{E}\left[\mathrm{P}(y \mid \sigma, \mathbf{z}) - \mathrm{P}(y \mid \bar{\sigma}, \mathbf{z})\right] \\ &= e(\sigma) \, . \end{aligned}$$

□

Exceptional cases aside, in practice, we often do not know the complete causal graph, and hence we do not know if we are considering—or have even measured—complete $\mathbb{Z}$. In an attempt to block any path other than the direct ones between $\mathbb{X}$ and Y, a naive approach would be to include as many variables in $\mathbb{Z}$ as possible. In addition to potentially violating Def. 6.2.1, the empirical estimation of the effect gets harder with more control variables, which we will discuss next.

One strategy would be to initialise $\mathbb{Z}$ to the best of our knowledge, which could be the empty set, and then discover the rules with the strongest observational effect. From these, we can then carefully select those that we wish to add to $\mathbb{Z}$, and then iterate to investigate whether there exist strong observational effect rules between $\mathbb{X}$ and Y conditioned on $\mathbb{Z}$. While this does not guarantee we discover the true $\mathbb{Z}$, it does provide a natural approach to causal exploration—as well as to iterative data mining, where we wish to discover hypotheses that explain the data beyond what we already know (Hanhijärvi et al., 2009).

6.2.2 Statistical Considerations

In practice, we want to estimate $e(\sigma)$ from a sample drawn from the population. Suppose that we have a sample of N instances **stratified** by $\mathbb{Z}$ from the population (or in practice, the sample size is large enough to give relatively accurate estimates of the marginal distribution of $\mathbb{Z}$). The naive estimator of the observational effect $e(\sigma)$ is the estimator based on the empirical distribution $\hat{\mathrm{P}}$, i.e. the **plug-in** estimator:

$$\begin{aligned} \widehat{e}(\sigma) &= \mathbb{E}\left[\hat{\mathrm{P}}(y \mid \sigma, \mathbf{z}) - \hat{\mathrm{P}}(y \mid \bar{\sigma}, \mathbf{z})\right] \\ &= \sum_{\mathbf{z} \in \mathscr{Z}} \Big(\hat{\mathrm{P}}(y \mid \sigma, \mathbf{z}) - \hat{\mathrm{P}}(y \mid \bar{\sigma}, \mathbf{z})\Big) \hat{\mathrm{P}}(\mathbf{z}) \\ &= \sum_{\mathbf{z} \in \mathscr{Z}} (\hat{p}_{\sigma, \mathbf{z}} - \hat{p}_{\bar{\sigma}, \mathbf{z}}) \hat{\mathrm{P}}(\mathbf{z}) \, , \end{aligned}$$

where $\mathscr{Z}$ is the joint domain of $\mathbb{Z}$, $\hat{p}_{\sigma,\mathbf{z}} = \hat{\mathrm{P}}(y \mid \sigma, \mathbf{z})$, and $\hat{p}_{\bar{\sigma},\mathbf{z}} = \hat{\mathrm{P}}(y \mid \bar{\sigma}, \mathbf{z})$. In a stratified sample, $\hat{\mathrm{P}}(\mathbf{z})$ is the same as $\mathrm{P}(\mathbf{z})$ for all $\mathbf{z}$. As the empirical distribution is an unbiased estimator of the population distribution, the plug-in estimator is an **unbiased** estimator of the observational effect.

Although unbiased, the plug-in estimator shows high variance for rules with overly small sample sizes for either of the two events, σ or $\bar{\sigma}$. To illustrate this, in Fig. 6.3 (left), we show the score distribution for the plug-in estimator for a very specific rule of five conditions, and see that while it is close to the true observational effect, it shows very high variance in small samples. This high variance is problematic, as it leads to overfitting: if we use this estimator for the optimisation task over a very large space of rules, the variance will turn into a strong positive bias—we will overestimate the effects of rules from the sample—that dominates the search, and we end up with random results of either extremely specific or extremely general rules.

We address this problem of high variance by biasing the plug-in estimator. In particular, we introduce bias in terms of our confidence in the point estimates using confidence intervals. Note that we need not quantify the confidence of the point estimate $\hat{\mathrm{P}}(\mathbf{z})$ as $\hat{\mathrm{P}}(\mathbf{z}) = \mathrm{P}(\mathbf{z})$; the point estimates of concern are the conditional probabilities $\hat{p}_{\sigma,\mathbf{z}}$ and $\hat{p}_{\bar{\sigma},\mathbf{z}}$.

In repeated random samples of instances with $\sigma = \top$ and $\mathbb{Z} = \mathbf{z}$ from the population, the number of instances with *successful* outcome y is a binomial random variable with the success probability $\mathrm{P}(y \mid \sigma, \mathbf{z})$. In a $\mathbf{z}$ stratum of $\mathbb{Z}$, let $n_{\sigma,\mathbf{z}}$ and $n_{\bar{\sigma},\mathbf{z}}$ be the number of instances that satisfy σ and $\bar{\sigma}$, respectively. Then the one-sided binomial confidence interval of $\hat{p}_{\sigma,\mathbf{z}}$, using a normal approximation of the error distribution, is given by $\beta\sqrt{\hat{p}_{\sigma,\mathbf{z}}(1-\hat{p}_{\sigma,\mathbf{z}})/n_{\sigma,\mathbf{z}}}$, where β is the $1-\alpha/2$ quantile of a standard normal distribution for an error rate α. For a 95% confidence level, for instance, the error rate is $\alpha = 0.05$, thereby $\beta = 1.96$. We can easily verify that the maximum value of $\hat{p}_{\sigma,\mathbf{z}}(1-\hat{p}_{\sigma,\mathbf{z}})$ is $1/4$, and hence the maximum value of the one-sided confidence interval is $\beta/(2\sqrt{n_{\sigma,\mathbf{z}}})$. Taking a conservative approach, we bias the difference $\hat{p}_{\sigma,\mathbf{z}} - \hat{p}_{\bar{\sigma},\mathbf{z}}$ by subtracting the sum of the maximum values of the one-sided confidence intervals of the point estimates, this results in

$$\tau(\mathbf{z}) = (\hat{p}_{\sigma,\mathbf{z}} - \hat{p}_{\bar{\sigma},\mathbf{z}}) - \left(\beta/(2\sqrt{n_{\sigma,\mathbf{z}}}) + \beta/(2\sqrt{n_{\bar{\sigma},\mathbf{z}}})\right) .$$

Note that $\tau(\mathbf{z})$ lower bounds the true probability mass difference in the population with confidence $1-\alpha$. That is, there is a $1-\alpha$ chance that the true difference is larger than $\tau(\mathbf{z})$. For a fixed β, the lower bound gets tighter with increasing sample size. In fact, it is easy to see that $\tau(\mathbf{z})$ is a consistent estimator of the true probability mass difference in the population; the bias term vanishes asymptotically. More formally, for a fixed finite β, we have

$$\lim_{\min(n_{\sigma,\mathbf{z}}, n_{\bar{\sigma},\mathbf{z}}) \to \infty} \beta/(2\sqrt{n_{\sigma,\mathbf{z}}}) + \beta/(2\sqrt{n_{\bar{\sigma},\mathbf{z}}}) = 0 .$$

As we deal with empirical probabilities, we can express $\tau(\mathbf{z})$ in terms of counts in a contingency table. Suppose that we have a contingency table as shown in Tab. 6.1 (left) for a $\mathbf{z}$ stratum. Then we can express $\tau(\mathbf{z})$ in terms of the cell counts in the contingency table as

$$\tau(\mathbf{z}) = \frac{a}{n_{\sigma,\mathbf{z}}} - \frac{c}{n_{\bar{\sigma},\mathbf{z}}} - \frac{\beta}{2\sqrt{n_{\sigma,\mathbf{z}}}} - \frac{\beta}{2\sqrt{n_{\bar{\sigma},\mathbf{z}}}} .$$

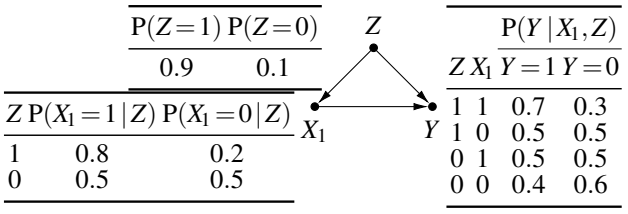


FIGURE 6.2: A toy causal graph of three variables X_1, Y and Z.

In the extreme case, however, a rule may select all or none of the instances in a stratum, resulting in $n_{\sigma,\mathbf{z}} = 0$ or $n_{\bar{\sigma},\mathbf{z}} = 0$, and hence the empirical conditional probability mass functions can be undefined. In practice, we encounter this problem often, both due to specificity of a rule as well as small sample sizes to begin with.

As a remedy, we apply the Laplace correction to the score. That is, we increment count of each cell in the contingency table by one. This way we start with a uniform distribution within each stratum of $\mathbf{Z}$. Hence a stratum of size n increases to $n+4$, and the total effective sample size increases from N to $N+4|\mathscr{Z}|$. After applying Laplace correction, we have $\hat{\mathrm{P}}(\mathbf{z}) = (n+4)/(N+4|\mathscr{Z}|)$, and $\tau(\mathbf{z})$ is given by

$$\tau(\mathbf{z}) = \frac{a+1}{n_{\sigma,\mathbf{z}}+2} - \frac{c+1}{n_{\bar{\sigma},\mathbf{z}}+2} - \frac{\beta}{2\sqrt{n_{\sigma,\mathbf{z}}+2}} - \frac{\beta}{2\sqrt{n_{\bar{\sigma},\mathbf{z}}+2}} \,.$$

After introducing the bias and applying the Laplace correction to the plug-in estimator, we obtain the **reliable** estimator of the observational effect as

$$\widehat{r}(\sigma) = \sum_{\mathbf{z}\in\mathscr{Z}} \tau(\mathbf{z})\hat{\mathrm{P}}(\mathbf{z}) \,. \tag{6.2}$$

Although biased, $\widehat{r}(\sigma)$ is still a **consistent** estimator of the observational effect. Importantly, in contrast to the plug-in estimator, the reliable estimator is much better at generalisation as it avoids overfitting.

Consider the following example to see the generalisation behaviour of the estimators. Suppose that we generate the population using the causal graph in Fig. 6.2. In addition, we generate five uniformly distributed binary actionable variables, $X_2, X_3, \ldots, X_6$ that are independent of each other as well as the rest of the variables. We can now numerically estimate the variance of the two estimators for a specific rule, e.g. $\sigma \equiv X_1 = 1 \wedge X_2 = 0 \wedge X_3 = 1 \wedge X_4 = 1 \wedge X_5 = 0 \wedge X_6 = 0$, which does not only contain causal variable X_1 but also five actionable variables that are independent of the target Y.

To do so, we draw stratified samples of increasing sizes from the population, and report $\widehat{e}(\sigma)$ and $\widehat{r}(\sigma)$ scores averaged over 25 simulations along with one sample standard deviation in Fig. 6.3 (left). We observe that variances of both estimators decrease with increasing sample size. Although the reliable estimator is biased, its variance is relatively low compared to the plug-in estimator. As a result of this low variance, unlike the plug-in estimator, the reliable estimator is indeed able to avoid overfitting, and hence, better at generalisation. Let σ^* denote the top-1 rule in the population, i.e. $\sigma^* = \mathrm{argmax}_{\sigma\in\mathscr{L}}\, e(\sigma)$. Let φ^* denote the top-1 rule using the plug-in estimator, i.e. $\varphi^* = \mathrm{argmax}_{\sigma\in\mathscr{L}}\, \widehat{e}(\sigma)$, and ρ^* denote the top-1 rule using the reliable estimator, i.e. $\rho^* = \mathrm{argmax}_{\sigma\in\mathscr{L}}\, \widehat{r}(\sigma)$. In Fig. 6.3 (right), we plot $e(\varphi^*)$ against $e(\rho^*)$. We observe that with increasing sample sizes $e(\rho^*)$ is

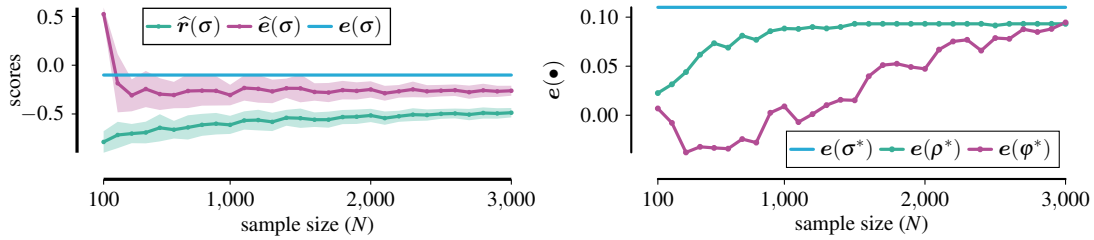


FIGURE 6.3: From the population generated using the causal graph of Fig. 6.2 together with 5 additional independent random actionable variables $X_2, \ldots, X_6$, we show (left) variance of the plug-in and reliable estimator of the observational effect for a specific rule that contains variables that are independent of the target, and (right) generalisation error of the effect estimators.

both relatively closer, as well as converges much faster to the reference $e(\sigma^*)$, which is in agreement with both theory and intuition.

6.3 Discovering Rules

Now that we have a reliable and consistent score for the observational effect, we turn to the problem of discovering rules that yield maximal reliable observational effect. Below, we provide the formal problem definition.

Definition 6.3.1 (Top-k causal rule discovery)**.** *Given a sample and a positive integer k, find a set $\mathscr{F}_k \subseteq \mathscr{L}$, $|\mathscr{F}_k| = k$, such that for all $\sigma \in \mathscr{F}_k$ and $\varphi \in \mathscr{L} \setminus \mathscr{F}_k$, $\widehat{r}(\sigma) \geq \widehat{r}(\varphi)$.*

Given the hardness of empirical effect maximisation problems (Wang et al., 2005), it is unlikely that the optimisation of the reliable observational effect allows a worst-case polynomial algorithm. While the exact computational complexity of the causal rule discovery problem is open, here we proceed to develop a practically efficient algorithm using the branch-and-bound paradigm.

6.3.1 Branch-and-Bound Search

The branch-and-bound search scheme (Mehlhorn and Sanders, 2008) finds a solution that optimises the objective function $f : \Omega \to \mathbb{R}$, among a set of admissible solutions Ω, also called the search space. Let $\text{ext}(\sigma)$, also called the extension of σ, denote the subset of instances in the sample that satisfy σ The generic search scheme for a branch-and-bound algorithm requires the following two ingredients:

- A **refinement operator** $b : \mathscr{L} \to \mathscr{P}(\mathscr{L})$ that is monotone, i.e. for $\sigma, \varphi \in \mathscr{L}$ with $\varphi = b(\sigma)$ it holds that $\text{ext}(\varphi) \subseteq \text{ext}(\sigma)$, and that non-redundantly generates the search space $\mathscr{L}$. That is, for every rule $\sigma \in \mathscr{L}$, there is a unique sequence of rules $\sigma_0, \sigma_1, \ldots, \sigma_\ell = \sigma$ with $\sigma_i = b(\sigma_{i-1})$.
- An **optimistic estimator** $\tilde{f} : \Omega \to \mathbb{R}$ that provides an upper bound on the objective function attainable by extending the current rule to more specific rules. That is, it holds that $\tilde{f}(\sigma) \geq f(\varphi)$ for all $\varphi \in \mathscr{L}$ with $\text{ext}(\varphi) \subseteq \text{ext}(\sigma)$.

A branch-and-bound algorithm simply enumerates the search space $\mathscr{L}$ starting from the root ϕ using the refinement operator $\boldsymbol{b}$ (branch), but based on the optimistic estimator $\tilde{f}$ prunes those branches that cannot yield improvement over the best rules found so far (bound).

The optimistic estimator depends on the objective function, and there are many optimistic estimators for an objective function f. Not all of these are equally well-suited in practice, as the tightness of the optimistic estimator determines its pruning potential. We consider the **tight optimistic estimator** (Grosskreutz et al., 2008) given by

$$\begin{aligned}\tilde{f}(\sigma) &= \max\{f(Q) \mid Q \subseteq \mathrm{ext}(\sigma)\} \\ &\geq \max\{f(\varphi) \mid \mathrm{ext}(\varphi) \subseteq \mathrm{ext}(\sigma) \text{ for all } \varphi \in \mathscr{L}\} .\end{aligned}$$

The branch-and-bound search scheme also provides an option to trade-off the optimality of the result for the speed. Instead of asking for the f-optimal result, we can ask for the γ-approximation result for some approximation factor $\gamma \in (0,1]$. This is done by relaxing the optimistic estimator, i.e. $\tilde{f}(\sigma) \geq \gamma f(\varphi)$ for all $\varphi \in \mathscr{L}$ with $\mathrm{ext}(\varphi) \subseteq \mathrm{ext}(\sigma)$. Lower γ generally yields better pruning, at the expense of guarantees on the quality of the solution.

In our problem setting, we can define the refinement operator based on the lexicographical ordering of propositions:

$$\boldsymbol{b}(\sigma) = \{\sigma \wedge \pi_i \mid \pi_i \in \boldsymbol{\pi}, i > \max\{j : \pi_j \in \boldsymbol{\pi}^{(\sigma)}\}\} ,$$

where $\boldsymbol{\pi}$ is the set of propositions and $\boldsymbol{\pi}^{(\sigma)}$ is the subset of $\boldsymbol{\pi}$ used in σ. In practice, we need more sophisticated refinement operators in order to avoid the inefficiency resulting from a combinatorial explosion of equivalent rules. This, we can do by defining a closure operator on the rule language (see, e.g. Boley and Grosskreutz (2009)), which we also employ in our experimental evaluation. Next we derive an optimistic estimator for the objective function $\widehat{r}$.

6.3.2 Efficient optimistic estimator

If we look at the definition of $\widehat{r}(\sigma)$ in Eq. (6.2), we see that, regardless of σ, $\hat{\mathrm{P}}(\mathbf{z})$ remains the same for a $\mathbf{z}$ stratum. Thus, we can obtain an optimistic estimator of $\widehat{r}(\sigma)$ by simply bounding $\tau(\mathbf{z})$ for each $\mathbf{z}$ stratum. Let $\tilde{\tau}(\mathbf{z})$ denote the optimistic estimator of $\tau(\mathbf{z})$. Then the optimistic estimator of $\widehat{r}(\sigma)$ is given by

$$\tilde{r}(\sigma) = \sum_{\mathbf{z} \in \mathscr{Z}} \tilde{\tau}(\mathbf{z}) \hat{\mathrm{P}}(\mathbf{z}) .$$

To derive the optimistic estimator $\tilde{\tau}(\mathbf{z})$, for clarity of exposition we first project $\tau(\mathbf{z})$ in terms of free variables a and b, such that we can write

$$\tau(a,b) = \frac{a+1}{a+b+2} - \frac{n_1 - a + 1}{n - a - b + 2} - \frac{0.5\beta}{\sqrt{a+b+2}} - \frac{0.5\beta}{\sqrt{n-a-b+2}} .$$

Suppose that we have a contingency table as shown in Tab. 6.1 (left) for a $\mathbf{z}$ stratum with the rule σ. The refinement of σ, $\sigma' = \boldsymbol{b}(\sigma)$, results in a contingency table as shown in Tab. 6.1 (right). Note that n_1, n_0, and n do not change within a $\mathbf{z}$ stratum regardless of the rule. Since $\mathrm{ext}(\sigma') \subseteq \mathrm{ext}(\sigma)$ holds for any $\sigma' = \boldsymbol{b}(\sigma)$, we have the following relations: $a' \leq a$ and $b' \leq b$.

Table 6.1: Contingency tables for (left) a rule σ, and (right) its refinement $\sigma' = b(\sigma)$ for a $\mathbf{z}$ stratum of $\mathbb{Z}$.

	$Y=y$	$Y\neq y$	
$\sigma=\top$	a	b	
$\sigma=\bot$	c	d	
$\sum$	n_1	n_0	n

	$Y=y$	$Y\neq y$	
$\sigma'=\top$	a'	b'	
$\sigma'=\bot$	c'	a'	
$\sum$	n_1	n_0	n

This implies that the subsets of the extensions of σ will have contingency table counts a' in the range $\{0,1,\ldots,a\}$, and b' in the range $\{0,1,\ldots,b\}$. Let $\mathscr{C} = \{0,1,\ldots,a\} \times \{0,1,\ldots,b\}$. Then the optimistic estimator of $\tau(\mathbf{z})$ can be defined in terms of $\mathscr{C}$ as

$$\tilde{\tau}(\mathbf{z}) \geq \max_{(a',b')\in\mathscr{C}} \tau(a',b') \,.$$

This shows that we have the optimistic estimate of $\tau(\mathbf{z})$ by simply taking the maximum value of τ from all possible configurations $\mathscr{C}$. In the following proposition, we show that we can have a **tight optimistic estimator** that can be computed in time linear to the number of observations in the contingency table.

Proposition 1. *Let $\mathscr{C} = \{0,1,\ldots,a\} \times \{0,1,\ldots,b\}$ be the set of all possible configurations of (a',b') in Tab. 6.1 (right) that can result from refinements of a rule σ from the contingency table of Tab. 6.1 (left). Then the **tight optimistic estimator** of $\tau(\mathbf{z})$ is given by*

$$\tilde{\tau}_t(\sigma,\mathbf{z}) = \max_{a'\in\{0,1,\ldots,a\}} \frac{a'+1}{a'+2} - \frac{n_1-a'+1}{n-a'+2} - \frac{\beta}{2\sqrt{a'+2}} - \frac{\beta}{2\sqrt{n-a'+2}}.$$

Proof. The expression for $\tau(a',b')$ from the contingency table in Tab. 6.1 (right) is given by

$$\tau(a',b') = \frac{a'+1}{a'+b'+2} - \frac{n_1-a'+1}{n-a'-b'+2} - \frac{\beta}{2\sqrt{a'+b'+2}} - \frac{\beta}{2\sqrt{n-a'-b'+2}}.$$

Combining the first and the third term above, we get

$$\lambda_z(a',b') = \frac{2a'+2-\beta\sqrt{a'+b'+2}}{2(a'+b'+2)} - \frac{n_1-a'+1}{n-a'-b'+2} - \frac{\beta}{2\sqrt{n-a'-b'+2}}.$$

Note that if we fix the value of a', then the value of b' that maximises $\tau(a',b')$ has to maximise the first term above, but minimise the other two terms. Observe that $b'=0$, out of $b' \in \{0,1,\ldots,b\}$, does both simultaneously. Thus we have the following relation: $\tau(a',0) > \tau(a',b')$ for all $b' > 0$.

The tight optimistic estimator of $\tau(\mathbf{z})$ is the maximum value over all possible configurations $\mathscr{C}$ given by

$$\begin{aligned}\tilde{\tau}_t(\sigma,\mathbf{z}) &= \max_{a'\in\{0,1,\ldots,a\}} \tau(a',0) \\ &= \max_{a'\in\{0,1,\ldots,a\}} \frac{a'+1}{a'+2} - \frac{n_1-a'+1}{n-a'+2} - \frac{\beta}{2\sqrt{a'+2}} - \frac{\beta}{2\sqrt{n-a'+2}}.\end{aligned}$$

□

Tight optimistic estimators give us very high pruning power at much lower computational complexity than the strictest optimistic estimator, but, is still not free: we have to compute it for every node in our search tree, and although it only has a linear time complexity, for large search spaces this may be costly. It is therefore an interesting question to ask whether, at the expense of tightness, we can obtain an optimistic estimator with a closed form expression, one that we can hence compute in constant time. It turns out that based on our tight optimistic estimator we achieve so with relative ease.

Proposition 2. *Let $\mathscr{C} = \{0,1,\ldots,a\} \times \{0,1,\ldots,b\}$ be the set of all possible configurations of (a',b') in Tab. 6.1 (right) that can result from the refinement of a rule σ from the contingency table of Tab. 6.1 (left). Then the closed-form optimistic estimator of $\tau(\mathbf{z})$ is given by*

$$\tilde{\tau}_l(\sigma,\mathbf{z}) = \frac{a+1}{a+2} - \frac{n_1-a+1}{n-a+2} - \frac{\beta}{2\sqrt{a+2}}.$$

Proof. From Proposition 1, we have the tight optimistic estimator of $\tau(\mathbf{z})$ as

$$\begin{aligned}\tilde{\tau}_t(\sigma,\mathbf{z}) &= \max_{a'\in\{0,1,\ldots,a\}} \frac{a'+1}{a'+2} - \frac{n_1-a'+1}{n-a'+2} - \frac{\beta}{2\sqrt{a'+2}} - \frac{\beta}{2\sqrt{n-a'+2}} \\ &< \max_{a'\in\{0,1,\ldots,a\}} \frac{a'+1}{a'+2} - \frac{n_1-a'+1}{n-a'+2} - \frac{\beta}{2\sqrt{a'+2}}.\end{aligned}$$

Suppose that we have $\rho(a') = \frac{a'+1}{a'+2} - \frac{n_1-a'+1}{n-a'+2} - \frac{\beta}{2\sqrt{a'+2}}$. Then it holds that

$$\tilde{\tau}_t(\sigma,\mathbf{z}) \leq \max_{a'\in\{0,1,\ldots,a\}} \rho(a').$$

In words, the maximum value of ρ upper bounds the tight optimistic estimate $\tilde{\tau}_t(\sigma,\mathbf{z})$. Hence the maximum of ρ is also an optimistic estimator of $\tau(\mathbf{z})$, albeit a loose one. Next we show that ρ is monotonically increasing with a'. To this end, we relax ρ to a continuous domain. In particular, we consider $a' \in \mathbb{R}_{\geq 0}$, which naturally subsumes the discrete domain $\{0,1,\ldots,a\}$. Then we take the first derivative of ρ with respect to a'. Thus we have

$$\begin{aligned}\frac{d\rho}{da'} &= \frac{1}{(a'+2)} - \frac{(a'+1)}{(a'+2)^2} + \frac{1}{n-a'+2} - \frac{n_1-a'+1}{(n-a'+2)^2} + \frac{\beta}{4(a'+2)^{3/2}} \\ &= \frac{1}{(a'+2)^2} + \frac{n-n_1+1}{(n-a'+2)^2} + \frac{\beta}{4(a'+2)^{3/2}} \\ &> 0.\end{aligned}$$

Since the first derivative is positive, ρ is monotonically increasing w.r.t. a'. Therefore the maximum of ρ is at $a' = a$, which is given by

$$\tilde{\tau}_l(\sigma,\mathbf{z}) = \frac{a+1}{a+2} - \frac{n_1-a+1}{n-a+2} - \frac{\beta}{2\sqrt{a+2}}.$$

□

We now have two optimistic estimators for $\tau(\mathbf{z})$, a tight one, $\tilde{\tau}_t(\sigma,\mathbf{z})$ at a computational cost of $O(N)$, and a loose one, $\tilde{\tau}_l(\sigma,\mathbf{z})$ at $O(1)$. In the experiments we will evaluate the performance of both estimators.

6.4 Related Work

Rule discovery (Fürnkranz et al., 2012) is a well-studied topic within data mining, but relatively little work has been done from a causal perspective.

In rule-based classification the goal is to find a (set of) rules that together optimally predict the target label. Classic approaches include CN2 (Lavrač et al., 2004), and FOIL (Quinlan and Cameron-Jones, 1995). In more recent work, the attention shifted from accuracy to optimising more reliable scores, such as Area-Under-ROC (Fürnkranz and Flach, 2005). While related, the overall goal in learning classification rules is different than ours; we want to find rules that describe the strong causal effects, rather than separate two classes.

Association rules (Agrawal et al., 1993) are implications of the form $I \Rightarrow J$, where I and J are disjoint sets of items. The interestingness of an association rule is typically measured in terms of its relative occurrence frequency. To get reliable rules, we can impose hard constraints on the relative occurrence frequency of an association rule. Despite that, within this framework we conflate the goal of finding rules with large effect size with the relative occurrence frequency of the rule. Contrast patterns (Dong and Li, 1999; Dong and Bailey, 2012), otherwise known as emerging patterns, are patterns whose supports differ significantly between datasets. As the support of a pattern is an empirical effect measure, without special measures such as taken here, emerging patterns tend to overfit the given sample and hence capture unreliable statements that are not necessarily characteristic of the underlying domain.

Subgroup discovery (Wrobel, 1997; Friedman and Fisher, 1999) is a related, but subtly different task. Most subgroup discovery methods optimise a surrogate function based on some statistical null hypothesis test. The resulting objective functions are usually some multiplicative combination of coverage and effect and, hence, do not consistently optimise for large effect. Also patterns found through standard subgroup discovery frameworks do not correct for the influence of confounder and are hence purely associational. Closer to our approach is RAWR (Kalofolias et al., 2017), which discovers patterns that both have large deviation from the mean of the population, but at the same time are also representative with respect to a univariate binary control variable Z in terms of statistical parity. Besides that we introduce a reliable measure of effect, our framework allows for control variables of higher dimensionality that, under the specific circumstances, directly optimises the causal effect.

Silverstein et al. (2000) test for pairwise dependence and conditional independence relationships to discover causal associations rules that consist of a univariate antecedent given a univariate control variable; they do not find the conditions under which the effect is visible on the target. Li et al. (2015) are specifically concerned with discovering causal association rules from observational data given a target. They propose to do so by first mining association rules with the target as a consequent, and performing cohort studies per rule. Unlike our setup, they optimise the plug-in estimator of the odds ratio.

Atzmueller and Puppe (2009) propose a semi-automatic approach to discovering causal interactions by mining subgroups using a chosen quality function, inferring a causal network over these, and visually presenting this to the user. They discover causal relationships among the rules, and not rules that have causal effect on the target. Causal falling rule lists (Wang and Rudin, 2017) are sequences of "if-then" rules over the covariates such that the effect of a specific intervention on the target decreases monotonically down the list. Our formulation, on the other hand, is aimed at finding top-k interventions, represented by rules, that have

maximal effect on the target given the control variables.

Shamsinejadbabaki et al. (2013) discover rules σ for which $\mathrm{P}(Y = y \mid do(\sigma = \top))$ differs from $\mathrm{P}(Y = y \mid \sigma = \top)$ on a partial directed acyclic graph of all the variables ($\mathbf{X}$, Y, and $\mathbb{Z}$) using the plug-in estimator. First, their effect measure is different than ours. Second, such effect measure would not be able to find actions from a causal diagram with no spurious paths, say $X \rightarrow Y$, because intervention and observation are equivalent in those scenarios, i.e. $\mathrm{P}(Y \mid do(X = x)) = \mathrm{P}(Y \mid X = x)$ (Pearl, 2009, Chap. 3.5.1).

Overall, despite the importance of the problem, to the best of our knowledge there does not exist a generally applicable, theoretically well-founded, efficient approach to discovering reliable rules with strong causal effect from observational data.

6.5 Experiments

We implemented the branch-and-bound search with priority-queue in free and open source `realKD`[2] Java library, and provide the source code online.[3] All experiments were executed single threaded on Intel Xeon E5-2643 v3 machine with 256 GB memory running Linux. We report the results at $\beta = 2.0$, which corresponds to a 95.45% confidence level. The coverage of a rule is a fraction of individuals that belong to its extension, defined as $\mathrm{cvg}(\sigma) = |\mathrm{ext}(\sigma)|/N$. We search for optimal top$-k$ rules, i.e. $\gamma = 1.0$, unless stated otherwise.

6.5.1 Efficiency

First we assess efficiency of the branch-and-bound search with the optimistic estimators. To this end, we search for top-1 rules in all the standard classification datasets from the `KEEL` repository (Alcalá-Fdez et al., 2011).[4] For each dataset, we select the classification target as the target variable, and randomly select one of the attributes as the control variable. We binarise a nominal target variable by mapping one of its outcomes to the positive category ($Y = 1$), and the rest to the negative category ($Y = 0$). We discretise a continuous real-valued actionable variable into maximum 8 equi-frequent bins.

In Tab. 6.2, we provide a summary of the datasets along with efficiency results. For each dataset, we report the target variable (Y), the set of control variables ($\mathbb{Z}$), the sample size (N), the number of actionable variables ($|\mathbf{X}|$), and the approximation factor (γ) such that the branch-and-bound implementation with the loose optimistic estimator finishes within an hour. Further, for both optimistic estimators, we give the runtime in seconds, denoted $\tilde{\tau}_l$ and $\tilde{\tau}_t$ resp., followed by the speed-up factor ($\tilde{\tau}_l/\tilde{\tau}_t$), and the number of nodes expanded during the search, denoted n_l and n_t resp., followed by the node reduction factor (n_l/n_t). To highlight the performance differences between the two optimistic estimators at a finer granularity, we report at least two significant non-zero digits for the runtime.

For most datasets, we observe that the loose optimistic estimator is already fast enough, retrieving the optimal top-1 result within 30 minutes. In general, however, the tight optimistic

[2] `https://bitbucket.org/realKD/`

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

[4] As both `census` and `adult` datasets are from the same census data of the United States in 1994, we only consider one of them in our evaluation. Further, instead of the `titanic` dataset in the `KEEL` repository with only 3 attributes, we consider the `titanic` training set with 10 attributes and 1 target from the `Kaggle` prediction challenge.

Table 6.2: Summary of the datasets used for the empirical evaluation along with the efficiency results. For each dataset, we report the chosen target variable (Y), the chosen control variables ($\mathbf{Z}$), the sample size (N), the number of actionable variables ($|\mathbf{X}|$), the approximation factor (γ), followed by the runtime in seconds of the branch-and-bound implementation of reliable observational effect with the two optimistic estimators along with the speed-up factor. Further we also present the number of nodes expanded by the two optimistic estimators followed by the node reduction factor.

						Runtime (s)			#nodes		
Dataset	Y	$\mathbf{Z}$	N	$\|\mathbf{X}\|$	γ	$\tilde{\tau}_l$	$\tilde{\tau}_t$	$\tilde{\tau}_l/\tilde{\tau}_t$	n_l	n_t	n_l/n_t
adult	class	sex	48,842	13	0.8	2,097	**1,717**	1.2	316,659	**258,575**	1.2
australian	class	a4	690	13	1.0	625	**146**	4.3	4,091,928	**952,175**	4.3
automobile	output	engine-type	205	24	1.0	264	**1**	147.6	2,079,978	**15,167**	137.1
breast	class	age	286	8	1.0	**72**	78	0.9	774	**420**	1.8
car	acceptability	safety	1,728	5	1.0	0.024	**0.02**	1.2	36	**33**	1.1
chess	class	bkblk	3,196	35	1.0	1,392	**851**	1.6	2,806,450	**1,613,398**	1.7
connect-4	class	a1	67,557	61	0.3	1,681	**1,679**	1.0	149,969	**140,707**	1.1
crx	class	a1	690	14	1.0	46	**14**	3.2	328,371	**101,621**	3.2
fars	injury-severity	case-state	100,968	28	0.8	1,167	**724**	1.6	37,432	**22,328**	1.7
flare	class	prev24hour	1,066	10	1.0	0.03	**0.014**	2.1	86	**32**	2.7
german	customer	statusAndSex	1,000	19	1.0	23	**8**	2.8	119,331	**43,007**	2.8
housevotes	class	el-salvador-aid	435	15	1.0	0.009	**0.007**	1.3	70	**57**	1.2
kddcup	class	atr-6	494,020	40	0.99	55	**37**	1.5	219	219	1.0
kr-vs-k	game	white-king-col	28,056	5	1.0	32	**30**	1.0	7,593	**7,304**	1.0
lymphography	classes	changes-in-lym	148	17	1.0	0.35	**0.14**	2.5	4,009	**1,666**	2.4
mushroom	class	gill-size	8,124	21	1.0	0.37	**0.307**	1.2	252	**215**	1.2
nursery	class	social	12,690	7	1.0	**0.51**	0.66	0.8	348	**279**	1.2
post-operative	decision	l-core	90	7	1.0	0.03	**0.016**	1.9	500	**258**	1.9
splice	class	pos1	3,190	59	1.0	1.9	**1.03**	1.9	3,119	**1,855**	1.7
tic-tac-toe	class	topleft	958	8	1.0	0.24	**0.11**	2.1	800	**488**	1.6
titanic	survived	sex	891	9	1.0	5.3	**4.5**	1.2	29,591	**26,700**	1.1
zoo	type	aquatic	101	15	1.0	0.038	**0.009**	4.2	499	**96**	5.2

estimator is faster than the loose one as indicated by the speed-up factors. This observation is also supported by the fact that the node reduction factor is almost in the same range as the speed-up factor for most datasets. In case of small datasets, such as `nursery` and `breast`, where the number of expanded nodes are relatively small, the loose optimistic estimator is marginally better the tight one, however.

For datasets such as `kddcup`, although the number of actionable variables is quite large, the branch-and-bound implementation is much faster with both optimistic estimators. As `kddcup` contains many binary actionable variables that are extremely sparse, refinements of a rule using the predicates on such actionable variables will lead to one of the stratums without sufficient statistical evidence, and likely worse score than the current best. Both optimistic estimators are able to identify such cases, thereby pruning the search space.

Overall we observe that the branch-and-bound search with both optimistic estimators finishes within minutes in most datasets, taking up to an hour (or more) for few datasets.

6.5.2 Quality of Top-k Rules

To assess the quality of discovered rules we consider synthetic data with known ground truth. In particular, we compare reliable observational effect $\widehat{r}(\sigma)$ to observational effect $\widehat{e}(\sigma)$ with an empty $\mathbb{Z}$, and weighted relative accuracy (Lavrač et al., 1999). In our case, the weighted relative accuracy of an event σ for an outcome y at the population level is given by

$$w(\sigma) = \mathrm{P}(\sigma)\Big(\mathrm{P}(y \mid \sigma) - \mathrm{P}(y)\Big) .$$

We apply the Laplace correction to the plug-in estimators of both observational effect and weighted relative accuracy. To generate synthetic data, we consider the toy causal graph from Fig. 6.2. First we generate the population using the joint distribution of the nodes in the causal graph. Then we add five other uniformly distributed binary actionable variables that are independent of each other as well as the rest of the variables. As only one actionable variable (X_1) affects the target Y in the causal graph, we expect the top-1 rule to contain only a proposition on that actionable variable, i.e. either $\sigma \equiv X_1 = 0$ or $\sigma \equiv X_1 = 1$ is a relevant result.

To evaluate the quality of the results, we use precision at k, which is the fraction of k rules that are relevant. In Fig. 6.4, we report precision at $k = 1$ averaged over 100 samples for increasing sample sizes. We observe that all the effect measures show high precision at $k = 1$ with a sufficient sample size. On smaller sample sizes, however, both weighted relative accuracy and observational effect without control variables lag behind reliable observational effect. These results, together with the results from Fig. 6.3, demonstrate that by taking into account both the structural and statistical problems, reliable observational effect measure discovers rules that are causal even on small sample sizes.

6.5.3 Qualitative Study on Real-World Data

Next we investigate whether rules discovered by reliable observational effect are meaningful. To this end, we consider two real-world datasets. Moreover, we show how the proposed framework can be used for mining rules iteratively.

Titanic For the first qualitative study, we consider the `titanic` training set from `Kaggle`.[5] The sinking of RMS Titanic is one of the notorious shipwrecks in history. One of

[5] `https://www.kaggle.com/c/titanic/data`

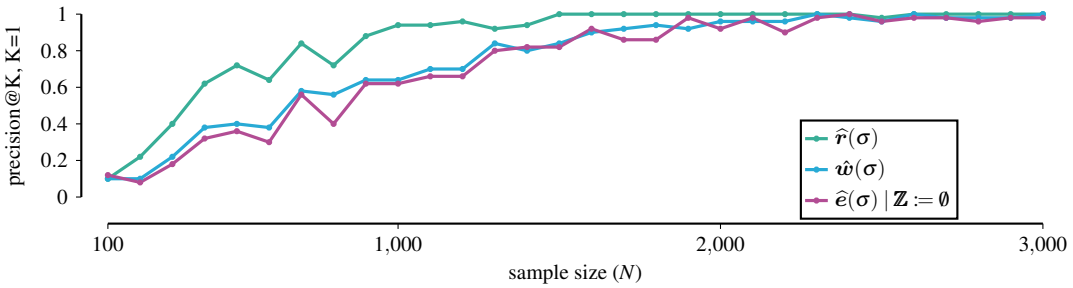


FIGURE 6.4: We show precision at $k = 1$, averaged over 100 samples, for the Laplace-corrected plug-in estimator of observational effect $\widehat{e}(\sigma)$ with an empty $\mathbb{Z}$, the Laplace corrected plug-in estimator of weighted relative accuracy, $\widehat{w}(\sigma)$, and reliable observational effect, $\widehat{r}(\sigma)$, for various sample sizes.

the reasons behind such tragic loss of lives was the lack of lifeboats. During the evacuation, some passengers were treated differently than the others; some groups of people were, hence, more likely to survive than the others. Thus, it is of interest to find the conditions that have effect on the survival. The dataset contains demographics and travel attributes of the passengers on board. The target of interest is the survival of a passenger.

In Tab. 6.3, we present the results of iterative rule mining on this dataset. For every iteration, we report the control variables, and the top-3 rules along with their coverage, followed by $\widehat{r}(\sigma \mid \mathbb{Z})$ and $\widehat{e}(\sigma \mid \mathbb{Z})$ scores. We start without control variables in the first iteration. In the subsequent iterations, we put top-1 rules discovered from previous iterations in $\mathbb{Z}$.

In the first iteration, without any control variables, we observe that being a female passenger from the first, or the second class has the highest effect on survival with a score of $\widehat{r}(\sigma \mid \mathbb{Z}) = 0.576$. It is well-known that passengers from different classes were treated differently during evacuation. What is interesting is that although females were more likely to survive, this only applied to the females from the first and the second class; this is also corroborated by the fact that roughly half of the females from the third class did not survive the mishap compared to the only one-tenth from the other two classes combined.

In the second iteration, the top-1 rule discovered in the first iteration is used as the control variable. Thus, controlling for the female passengers from the first and the second class, we find that children with fewer siblings, and parents on board have highest effect on survival. The fact that this rule came out on top with a coverage of only 4.6% demonstrates that reliable observational effect can discover rare rules.

In the third iteration, we control for the top-1 rules discovered in the previous two iterations, that is $\mathbb{Z} = \{\sigma_1, \sigma_2\}$, and find that being an unmarried female despite paying a low fare has the highest effect on survival with a score of $\widehat{r}(\sigma \mid \mathbb{Z}) = 0.003$. In the fourth iteration, where $\mathbb{Z} = \{\sigma_1, \sigma_2, \sigma_3\}$ we find that all top-3 rules have negative $\widehat{r}(\sigma \mid \mathbb{Z})$ scores. Although the $\widehat{e}(\sigma \mid \mathbb{Z})$ scores is positive for the top-1 rule, the negative $\widehat{r}(\sigma \mid \mathbb{Z})$ score indicates the lack of evidence for the rule. Therefore we stop after the fourth iteration.

Adult For the second qualitative study, we consider the `adult` dataset from the `KEEL` repository. The `adult` dataset contains socio-economic records of the population from the census database of the United States in 1994. The target of interest is the annual gross income ($>$ 50K or $\leq$ 50K). We ignore two attributes namely "Fnlwgt" which represents

Table 6.3: Results of iterative rule mining on the `titanic` dataset with "survival" as the target. We start without control variables in the first iteration. In the subsequent iterations, we control for the top-1 rules from previous iterations. "par-ch" stands for the number of parents/children aboard, and "sib-sp" for the number of siblings/spouses aboard.

Itr.	$\mathbb{Z}$	Top-3 rules (σ)	cvg(σ)	$\widehat{r}(\sigma \mid \mathbb{Z})$	$\widehat{e}(\sigma \mid \mathbb{Z})$
1	$\emptyset$	(σ_1) class $\leq 2 \wedge$ sex = female	0.1907	0.576	0.690
		class $\leq 2 \wedge$ sex = female $\wedge$ par-ch ≤ 2	0.1885	0.573	0.687
		class $\leq 2 \wedge$ sex = female $\wedge$ sib-sp ≤ 2	0.1874	0.572	0.686
2	$\{\sigma_1\}$	(σ_2) age $< 12.5 \wedge$ sib-sp $\leq 2 \wedge$ par-ch ≥ 1	0.0460	0.239	0.482
		age $< 12.5 \wedge$ sib-sp ≤ 2	0.0490	0.235	0.472
		age $< 12.5 \wedge$ sib-sp $\leq 2 \wedge$ par-ch $\geq 1 \wedge$ fare ≤ 65.8	0.0410	0.233	0.485
3	$\{\sigma_1, \sigma_2\}$	(σ_3) fare $< 19.85 \wedge$ title = Miss	0.1036	0.003	0.222
		title = Miss	0.2044	0.002	0.195
		fare $\leq 99.96 \wedge$ title=Miss	0.1750	-0.001	0.192
4	$\{\sigma_1, \sigma_2, \sigma_3\}$	(σ_4) embarked=C	0.1821	-0.036	0.149
		embarked=S	0.7760	-0.293	-0.117
		embarked=Q	0.0392	-0.369	-0.049

the number of people that the census takers believe that the observation represents, and "Education-num" which represents the highest level of education (edu) in numerical form.

In the first iteration, without any control variable, despite a mere 0.31% coverage, we observe the highest effect for people who hold a degree from a professional school; work for more than 30.5 hours per week; are older than mid-thirties; are married with a civilian spouse; and whose annual capital gain—profits from selling capital assets such as real estate, stocks—is positive.

In the second iteration, we find the highest effect for husbands in the households who are native to the US with a professional speciality, and incur an annual capital loss of no less than 184 USD. People who are specialists in their profession are likely to have a good living, and have money to invest in capital assets; although some of the assets can result in a net loss. Therefore a native US citizen with such socio-economic status is likely to earn more than 50K per annum in 1994 than the others.

In the third iteration, we find that being a self employed and incorporated white male has the highest effect on annual gross income. In the fourth iteration, we cannot find any rule with a positive $\widehat{r}(\sigma \mid \mathbb{Z})$ score; hence we stop.

Table 6.4: Results of iterative rule mining on the `adult` dataset with "annual gross income" ($>$ 50K or $\leq$ 50K) as the target. We start without control variables in the first iteration. In the subsequent iterations, we control for the top-1 rules discovered in the previous iterations. "hpw" stands for hours per week.

Itr.	$\mathbb{Z}$		Top-3 rules (σ)	cvg(σ)	$\widehat{r}(\sigma \mid \mathbb{Z})$	$\widehat{e}(\sigma \mid \mathbb{Z})$
1	$\emptyset$	(σ_1)	edu=prof-school $\wedge$ cap-gain $>$ 0.0 $\wedge$ age $\geq$ 34.5 $\wedge$ hpw $\geq$ 30.5 $\wedge$ mstatus=married-civ	0.0031	0.659	0.747
			occup=exec-manag. $\wedge$ cap-gain $>$ 0.0 $\wedge$ age $\geq$ 34.5 $\wedge$ hpw $>$ 47.5 $\wedge$ mstatus=married-civ $\wedge$ native=US	0.0049	0.658	0.729
			edu=prof-school $\wedge$ cap-gain $>$ 0.0 $\wedge$ hpw $\geq$ 30.5 $\wedge$ mstatus=married-civ	0.0034	0.656	0.741
2	$\{\sigma_1\}$	(σ_2)	occup=prof $\wedge$ native=US $\wedge$ rel=husband $\wedge$ cap-loss $\geq$ 184	0.0044	0.586	0.661
			occup=prof $\wedge$ native=US $\wedge$ rel=husband $\wedge$ cap-loss $\geq$ 184 $\wedge$ race=white	0.0043	0.585	0.660
			occup=prof $\wedge$ race=white $\wedge$ rel=husband $\wedge$ cap-loss $\geq$ 184	0.0046	0.584	0.657
3	$\{\sigma_1, \sigma_2\}$	(σ_3)	race=white $\wedge$ sex=male $\wedge$ work=self-emp-inc	0.0286	0.318	0.350
			sex=male $\wedge$ work=self-emp-inc	0.0303	0.317	0.349
			race=white $\wedge$ work=self-emp-inc	0.0324	0.289	0.319
4	$\{\sigma_1, \sigma_2, \sigma_3\}$		-	1.0	-0.972	-0.260

6.6 Discussion

In this work, we identified the conditions under which causal rule discovery is possible. The experimental results show that the proposed algorithm is fast and reliably discovers relevant causal rules. The results on synthetic data corroborates the necessity for both statistical and structural considerations to reliably discover relevant rules from a sample that are consistent with data generation process. On real-world data, the reliable observational effect finds sensible rules.

Although these results are promising, we see many possibilities for future work. One such direction would be to extend this work for a continuous real-valued target. We compute the reliable observational effect by grouping the individuals based on their value of control variables. As the number of control variables increases, the domain $\mathscr{Z}$ grows larger. Consequently we require a very large sample size to compute the reliable observational effect

with sufficient confidence. Therefore, it would make an interesting future work to develop an effect measure, and the algorithm to gracefully handle a large set of control variables.

To verify that the rules we discover from data are causal, we have to look at the causal structure of input variables. In practice, we often do not know the exact underlying causal structure. Under strict assumptions, however, we can discover a partially directed causal graph directly from data (Ch. 5 Spirtes et al., 2000; Chickering, 2002; Pearl, 2009, Ch. 2.5). As causal graph discovery is computationally expensive, using our domain knowledge, we can often rule out certain variables for causal rule discovery. For instance, smoking causes tar deposits in a person's lungs, therefore either `smoking` or `tar deposits` can be in our set of actionable variables $\mathbf{X}$. If we were to include both in $\mathbf{X}$, we would violate the third condition for an admissible input to causal rule discovery.

In the hope of blocking any spurious path between any $X \in \mathbf{X}$ and Y by $\mathbb{Z}$, a naive approach would be to include as many variables in $\mathbb{Z}$ as possible. Doing so, however, not only potentially violates the criterion for an admissible input, but also invites other statistical problems detailed in Sec. 6.2.2. A rather safe strategy is to consider those variables in $\mathbb{Z}$—using our domain knowledge—that are potentially not affected by any intervention on the actionable variables, otherwise known as pre-treatment covariates. For instance, smoking does not affect a person's sex, whereas it may affect a person's blood pressure. It is, therefore, safe to include `sex` in $\mathbb{Z}$, but not `blood pressure` as that could violate the second condition for an admissible input to causal rule discovery.

Note that despite the best of our knowledge, we can hardly be sure that we have no unobserved confounders. It is, therefore, typical in causal inference literature to assume that we have measured all the confounders, also known as the causal sufficiency assumption (Spirtes et al., 2000, Ch. 3.2.2). Assuming causal sufficiency, and that $\mathbb{Z}$ blocks all the spurious paths between any $X \in \mathbf{X}$ and Y, the noisy-or model (Koller and Friedman, 2009, Ch. 5.4.1), widely used in the medical domain, with three layers—the bottom layer for the target, the middle layer for the actionable variables, and the top one for the control variables—satisfies the criterion for an admissible input to causal rule discovery.

If we are merely interested in discovering associations, we can simply run the proposed method with an empty set of control variables, i.e. $\mathbb{Z} = \emptyset$. Besides, any violation of Definition 6.2.1 implies that we discover associations. The choice of β is up to a user's discretion. It is quite common to report results at the 95% confidence level, thereby $\beta = 1.96$. It is easy to see that the 0% confidence interval corresponds to the plug-in estimator, and that for higher confidence values we naturally need larger amounts of evidence.

Instead of searching for everything significant, we are only after top-k rules. As such, we are maximising the reliable observational effect. We want to discover those k rules from the sample that are also top-k in the population. Although the generalisation error bound of the reliable observational effect remains an open problem, the generalisation error plot on the synthetic data in Fig. 6.3 indicates that the reliable observational effect performs quite well even without a correction for multiple hypothesis testing.

6.7 Conclusion

Traditional rule discovery methods struggle with the consistent detection of conditions that have a strong causal effect on an output variable. In this work, we presented a novel rule discovery approach based on reliably estimating the observational effect given the

value of potential confounders. We also identified the conditions under which causal rule discovery is possible. We then demonstrated that the corresponding score is a conservative and consistent estimator of the causal effect and derived an efficient algorithm that detects meaningful rules on real datasets. Moreover, and of particular importance for both causal and associational data exploration, we showed that the presented approach naturally allows for iterative rule discovery. Causal assumptions are not verifiable even in principle, without controlled experiment. It would thus make an engaging future work to study the sensitivity of causal rule discovery to violations of assumptions for the admissible input. As we only worked with discrete variables, one avenue for further research would be to extend this work to continuous real-valued target and control variables.

Software Artefacts

We implemented the branch-and-bound algorithm in Java. Please follows the steps below to get the implementation running.

Preparing the Executable

Use the jar file provided in `http://eda.mmci.uni-saarland.de/dice/`, or follow the following steps:

- Checkout the development branch of the realkd repository from `https://bitbucket.org/realKD/realkd/overview`.
- Prepare the jar files using maven from inside the root folder.
- The compiled SNAPSHOT jar will be in the "target" directory.

Preparing Datasets

The dataset must be in one of the following file formats.

- arff
- xarf (`https://bitbucket.org/realKD/realkd/wiki/model/data/xarf`)

Inferring reliable (causal) rules

To infer reliable (causal) rules from data, we have to provide a job description file in the json format. A sample job file is provided in `http://eda.mmci.uni-saarland.de/dice/`. In short, we provide information about the dataset(s) in the "workspaces" field, and the computations to carry out on those workspaces in the "computations" field. For the detail information about the "computations" field, please refer to the "subgroupDiscovery.html" file inside the "kdondoc" directory. To run the job, we simply provide the job description file as an argument in the command line.

```
$ java -jar realkd.jar sgd.json
```

The result of the computation will be stored in the "output" directory.

Conclusion 7

In this thesis, we developed various techniques to identify causal relations from observational discrete data. First, we showed various ways to instantiate the algorithmic independence of conditionals (AIC) for bivariate causal inference on different data settings—univariate i.i.d. pairs, univariate non-i.i.d. pairs, and multivariate i.i.d. pairs. Some techniques, in theory, can identify the causal graph from the joint distribution, some cannot. Importantly, regardless of the identifiability result, we saw that the proposed methods are reasonably accurate in both simulated and real-world settings. Next, we looked beyond bivariate pairs to causal *rules* to identify the conditions that are most effective on the target variable of interest. In particular, we gave an efficient branch-and-bound search algorithm to discover reliable causal rules directly from data. The contributions of this thesis can be summarised as follows:

- On a pair of univariate i.i.d. discrete variables, we instantiated the AIC through the refined version of the Minimum Description Length (MDL) principle, and proposed CISC. Although the formulation follows directly from the statistically sound approximation of Kolmogorov complexity through the MDL principle, there were no theoretical results on the identifiability. To have an identifiable method, we turned to discrete additive noise models (ANMs), which are "generally" identifiable from the joint distribution. In particular, we formulated ANMs in terms of Shannon entropy, and proposed ACID. With an information-theoretic formulation, we avoid explicit statistical hypothesis testing for independence. Moreover, we showed that the refined MDL code of data w.r.t. the parametric family of multinomial distributions can be used as an estimator of the Shannon entropy, and proposed CRISP. As such, we showed the connection between AIC and ANMs which are two widely used frameworks for bivariate causal inference. Extensive experiments on synthetic and real-world data showed that the proposed methods are highly accurate and recover the ground truth.
- As discrete variables can also be non-i.i.d., next we considered a pair of univariate discrete-valued time series, or event sequences. To this end, we built an information-theoretic causal inference framework based on the foundations of Granger causality, which has close connection to the AIC. In particular, we instantiated the framework by the sequential normalised maximum likelihood codes, which are robust to model misspecification in

the sense that they are minimax optimal with respect to a model class. The proposed method CUTE runs in linear time, and through the evaluation, we observed that it is highly accurate compared to the state-of-the-art.

- Often times, we are interested in knowing whether a group of variables cause another group of variables. Therefore, we considered causal inference on a pair of multivariate binary random variables next. We modelled the causal mechanism by a set of decision trees. As refined MDL codes are computationally hard to compute on complex model classes beyond the exponential family, we instantiated the AIC through the two-part MDL codes. Although not identifiable, experiments on synthetic data demonstrated that the resulting method ORIGO is robust to noise and dimensionality, and recovers the ground truth from real-world data.
- There are situations where saying that a group of variables cause a target variable of interest does not fully satisfy one's curiosity. For a domain expert, for instance, it is of interest to know the most effective conditions that cause the target. To identify those conditions directly from data, next we studied the problem of deriving rules or policies from observational data that, when enacted on a complex system, cause a desired outcome. To this end, we investigated conditions under which the observational effect (which is what we can estimate from observational data) is an unbiased estimator of the causal effect. Then we developed a reliable estimator for the observational effect that does not suffer from overfitting when using it to optimise over the rule language, and gave an effective algorithm for finding optimal rules with respect to this estimator.

Although these results are promising, there is a plenty of room for improvement. Experiments show that CISC, CUTE and ORIGO—all of which deal with a pair of variables—work well in practice despite the lack of identifiability results. Although identifiability results are on the population level, i.e. joint distribution, and all we can do in practice is estimate the joint distribution from a sample, it is important to have such results rather than not have them at all. It would make an engaging future work to include missing identifiability results for those methods.

It is standard in causal discovery to assume that we have measured all the common causes of the observed variables, otherwise known as causal sufficiency assumption. For bivariate causal inference, we assumed causal sufficiency—that is, there is no hidden common cause. Although this leads to fine theoretical results, it is a strong assumption after all. Therefore it would be a worthwhile goal to relax that assumption. In fact, there has been some research in this direction for continuous real-valued data with ANMs (Janzing et al., 2009; Schölkopf et al., 2016; Janzing and Schölkopf, 2017; Kaltenpoth and Vreeken, 2019). It would be interesting to see whether NML codes could be used to solve this for discrete data.

For the discovered rules to be causal, the input must be admissible (Definition 6.2.1). Although Criterion (a), (b) and (d) are fairly standard in the literature, Criterion (c) is something new and specific to rule discovery. Criterion (c) requires that there are no edges between actionable variables. Over a large group of actionable variables, this can be a strong assumption. The naive way to remove this assumption would be to include rest of the actionable variables in the set of control variables to block any spurious path between actionable variables in a rule and the target via rest of the actionable variables. By doing so, however, we may not only violate other criteria, but the search procedure also gets complicated. Therefore, one avenue for future work would be to relax this assumption.

Another engaging direction for the future would be to extend the rule discovery framework to continuous real-valued variables.

Note that we need controlled experiment to make absolute statements about cause and effect. Even that, in practice, requires a stringent experiment design. To make things worse, controlled experiments cannot answer our all causal questions. The only viable alternative then is to carry out an observational study. Causal inference from observational data, however, rests on unverifiable assumptions, such as that of causal sufficiency. It is therefore important to state those assumptions in a formal language (as in additive noise models), and take conclusions from observational study as a suggestion. In support of open science, all implementations of the methods we proposed in this thesis together with used datasets are made publicly available online.

Bibliography

R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, 1993.

J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2–3):255–287, 2011.

A. Antos and I. Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19(3-4):163–193, 2001.

M. Atzmueller and F. Puppe. A knowledge-intensive approach for semi-automatic causal subgroup discovery. In *Knowledge Discovery Enhanced with Semantic and Social Information*, pages 19–36. Springer, 2009.

P. Bloebaum, D. Janzing, T. Washio, S. Shimizu, and B. Schölkopf. Cause-effect inference by comparing regression errors. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 900–909. PMLR, 2018.

M. Boley and H. Grosskreutz. Non-redundant subgroup discovery using a closure system. In *Machine Learning and Knowledge Discovery in Databases*, pages 179–194. Springer, 2009.

K. Budhathoki and J. Vreeken. Causal inference by compression. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 41–50, 2016.

K. Budhathoki and J. Vreeken. MDL for causal inference on discrete data. In *2017 IEEE 17th International Conference on Data Mining (ICDM)*, pages 751–756, 2017.

K. Budhathoki and J. Vreeken. Accurate causal inference on discrete data. In *2018 IEEE 18th International Conference on Data Mining (ICDM)*, pages 881–886, 2018a.

K. Budhathoki and J. Vreeken. Causal inference on event sequences. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 55–63, 2018b.

K. Budhathoki and J. Vreeken. Origo: causal inference by compression. *Knowledge and Information Systems*, 56(2):285–307, 2018c.

K. Budhathoki, M. Boley, and J. Vreeken. Rule discovery for exploratory causal reasoning. In *NeurIPS 2018 workshop on Causal Learning*, pages 1–14, 2018.

R. Cai, J. Qiao, K. Zhang, Z. Zhang, and Z. Hao. Causal discovery from discrete data using hidden compact representation. In *Advances in Neural Information Processing Systems 31*, pages 2666–2674. Curran Associates, Inc., 2018.

G. J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. *J. ACM*, 16(3):407–422, 1969.

Y. Chen, G. Rangarajan, J. Feng, and M. Ding. Analyzing multiple nonlinear time series with extended granger causality. *Phys. Let. A*, 324(1):26–35, 2004.

Z. Chen, K. Zhang, and L. Chan. Nonlinear causal discovery for high dimensional data: A kernelized trace method. In *2013 IEEE 13th International Conference on Data Mining (ICDM)*, pages 1003–1008, 2013.

D. M. Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

T. Chu and C. Glymour. Search for additive nonlinear time series causal models. *Journal of Machine Learning Research*, 9:967–991, 2008.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience New York, 2006.

J.-C. Croizet and M. Dutrévis. Socioeconomic status and intelligence: Why test scores do not equal merit. *Journal of Poverty*, 8(3):91–107, 2004.

A. P. Dawid. Beware of the dag! In *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008*, volume 6 of *Proceedings of Machine Learning Research*, pages 59–86, 2010.

D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *PRSocA*, 400(1818):97–117, 1985.

G. Dong and J. Bailey. *Contrast Data Mining: Concepts, Algorithms, and Applications*. Chapman & Hall/CRC, 1st edition, 2012.

G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52, New York, NY, USA, 1999. ACM.

J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

J. Fürnkranz and P. A. Flach. ROC 'n' rule learning - towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.

J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of Rule Learning*. Cognitive Technologies. Springer, 2012.

P. Gács, J. Tromp, and P. M. B. Vitányi. Algorithmic statistics. *IEEE Trans. Information Theory*, 47(6):2443–2463, 2001.

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

H. Grosskreutz, S. Rüping, and S. Wrobel. Tight optimistic estimates for fast subgroup discovery. In *Machine Learning and Knowledge Discovery in Databases*, pages 440–456. Springer, 2008.

P. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

P. Grünwald and P. M. B. Vitányi. Algorithmic information theory. *CoRR*, abs/0809.2754, 2008.

S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. Tell me something I don't know: randomization strategies for iterative data mining. In *KDD*, pages 379–388. ACM, 2009.

M. A. Hernán and J. M. Robins. *Causal Inference*. Boca Raton: Chapman & Hall/CRC, forthcoming edition, 2019.

E. H. Hill and M. C. Giammatteo. Socio -economic status and its relationship to school achievement in the elementary school. *Elementary English*, 40(3):265–270, 1963.

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 21*, pages 689–696. Curran Associates, Inc., 2009.

Y. Huang and S. Kleinberg. Fast and accurate causal inference from time series data. In *Florida Artificial Intelligence Research Society Conference*, pages 49–54, 2015.

A. Hyvärinen, S. Shimizu, and P. O. Hoyer. Causal modelling combining instantaneous and lagged effects: An identifiable model based on non-Gaussianity. In *Proceedings of the 25th International Conference on Machine Learning*, pages 424–431. ACM, 2008.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic markov condition. *IEEE Trans. Information Theory*, 56(10):5168–5194, 2010.

D. Janzing and B. Schölkopf. Detecting confounding in multivariate linear models via spectral analysis. *Journal of Causal Inference*, 6(1), 2017.

D. Janzing, J. Peters, J. Mooij, and B. Schölkopf. Identifying confounders using additive noise models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 249–257, Arlington, Virginia, United States, 2009. AUAI Press.

D. Janzing, P. Hoyer, and B. Schölkopf. Telling cause from effect based on high-dimensional observations. In *Proceedings of the 27th International Conference on Machine Learning*, pages 479–486. International Machine Learning Society, 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. *AIJ*, 182-183:1–31, 2012.

J. Kalofolias, M. Boley, and J. Vreeken. Efficiently discovering locally exceptional yet globally representative subgroups. In *2017 IEEE International Conference on Data Mining*, pages 197–206, 2017.

D. Kaltenpoth and J. Vreeken. We are not your real parents: Telling causal from confounded by mdl. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, pages 199–207. SIAM, 2019.

M. Kocaoglu, A. G. Dimakis, S. Vishwanath, and B. Hassibi. Entropic causal inference. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1156–1162. AAAI Press, 2017.

D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

A. Kolmogorov. Three approaches to the quantitative definition of information. *Problemy Peredachi Informatsii*, 1(1):3–11, 1965.

P. Kontkanen and P. Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Inf. Process. Lett.*, 103(6):227–233, 2007.

W. Kotlowski and P. Grünwald. Sequential normalized maximum likelihood in log-loss prediction. In *2012 IEEE Information Theory Workshop*, pages 547–551, 2012.

S. Kpotufe, E. Sgouritsa, D. Janzing, and B. Schölkopf. Consistency of causal inference under the additive noise model. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 478–495. PMLR, 2014.

N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In *Inductive Logic Programming*, pages 174–185. Springer, 1999.

N. Lavrač, B. Kavsek, P. A. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

J. Lemeire and E. Dirkx. Causal models as minimal descriptions of multivariate systems. `http://parallel.vub.ac.be/~jan`, 2006.

J. Li, T. D. Le, L. Liu, J. Liu, Z. Jin, B. Sun, and S. Ma. From observational studies to causal rule mining. *ACM Trans. Intell. Syst. Technol.*, 7(2):14:1–14:27, 2015.

F. Liu and L. Chan. Causal inference on discrete data via estimating distance correlations. *Neural Computation*, 28(5):801–814, 2016.

A. Marx and J. Vreeken. Causal inference on multivariate and mixed-type data. In *Machine Learning and Knowledge Discovery in Databases*, pages 655–671. Springer International Publishing, 2019.

K. Mehlhorn and P. Sanders. *Algorithms and Data Structures: The Basic Toolbox*. Springer Publishing Company, Incorporated, 1 edition, 2008.

T. Mononen and P. Myllymäki. Computing the multinomial stochastic complexity in sub-linear time. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models*, pages 209–216, 2008.

J. Mooij, D. Janzing, J. Peters, and B. Schölkopf. Regression by dependence minimization and its application to causal inference in additive noise models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 745–752, New York, NY, USA, 2009. ACM.

H.-V. Nguyen, E. Müller, J. Vreeken, and K. Böhm. Unsupervised interaction-preserving discretization of multivariate data. *Data Mining and Knowledge Discovery*, 28(5-6): 1366–1397, 2014.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009.

J. Peters, D. Janzing, and B. Schölkopf. Identifying cause and effect on discrete data using additive noise models. In *JMLR Workshop and Conference Proceedings Volume 9: AISTATS 2010*, pages 597–604. JMLR, 2010.

J. Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems 26*, pages 154–162, 2013.

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*. Adaptive Computation and Machine Learning Series. The MIT Press, 2017a.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA, 2017b.

U. Pötter and H.-P. Blossfeld. Causal inference from series of events. *European Sociological Review*, 17(1):21–32, 2001.

J. R. Quinlan and R. M. Cameron-Jones. Induction of logic programs: FOIL and related systems. *New Generation Comput.*, 13(3&4):287–312, 1995.

C. J. Quinn, T. P. Coleman, N. Kiyavash, and N. G. Hatsopoulos. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Comp. Neurosci.*, 30(1):17–44, 2011.

H. Reichenbach. *The Direction of Time*. University of California Press, Berkeley, California, 1956.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(1):465–471, 1978.

J. Rissanen. Strong optimality of the normalized ml models as universal codes. *IEEE Transactions on Information Theory*, 47:1712–1717, 2000.

J. Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Trans. Information Theory*, 47(5):1712–1717, 2001.

J. Rissanen and T. Roos. Conditional nml universal models. In *2007 Information Theory and Applications Workshop*, pages 337–341, 2007.

J. Rissanen and M. Wax. Measures of mutual and causal dependence between two time series. *IEEE Transactions on Information Theory*, 33(4):598–601, 1987.

R. Scheines. An introduction to causal inference. In *Causality in Crisis? University of Notre Dame*, pages 185–200. Press, 1997.

B. Schölkopf, D. Hogg, D. Wang, D. Foreman-Mackey, D. Janzing, C. J. Simon-Gabriel, and J. Peters. Modeling confounding by half-sibling regression. *Proceedings of the National Academy of Science*, 113(27):7391–7398, 2016.

T. Schreiber. Measuring information transfer. *Phys. Rev. Lett.*, 85:461–464, 2000.

E. Sgouritsa, D. Janzing, P. Hennig, and B. Schölkopf. Inference of cause and effect with unsupervised inverse regression. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 847–855. PMLR, 2015.

P. Shamsinejadbabaki, M. Saraee, and H. Blockeel. Causality-based cost-effective action mining. *Intelligent Data Analysis*, 17(6):1075–1091, 2013.

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. *Problems of Information Transmission*, 23(3):175–186, 1987.

T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki. Factorized normalized maximum likelihood criterion for learning bayesian network structures. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models*, pages 257–264, 2008.

C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 4(2):163–192, 2000.

R. J. Solomonoff. A formal theory of inductive inference. part I, II. *Information and Control*, 7:1–22224–254, 1964.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

N. Tatti and J. Vreeken. Finding good itemsets by packing data. In *2008 Eighth IEEE International Conference on Data Mining*, pages 588–597, 2008.

N. Vereshchagin and P. Vitanyi. Kolmogorov's structure functions and model selection. *IEEE Trans. Information Theory*, 50(12):3265–3290, 2004.

J. Vreeken. Causal inference by direction of information. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 909–917, 2015.

F. Wang and C. Rudin. Causal falling rule lists. In *Proceedings of 2017 Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

L. Wang, H. Zhao, G. Dong, and J. Li. On the complexity of finding emerging patterns. *Theoretical Computer Science*, 335(1):15–27, 2005.

S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Principles of Data Mining and Knowledge Discovery*, pages 78–87. Springer Berlin Heidelberg, 1997.

W. Wu and N. Hatsopoulos. Evidence against a single coordinate system representation in the motor cortex. *Exp. Brain Res.*, 175(2):197–210, 2006.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 647–655. AUAI Press, 2009.

J. Zscheischler, D. Janzing, and K. Zhang. Testing whether linear equations are causal: A free probability theory approach. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 839–847. AUAI Press, 2011.

Index