Origo: Causal Inference by Compression

Kailash Budhathoki · Jilles Vreeken

Abstract Causal inference from observational data is one of the most fundamental problems in science. In general, the task is to tell whether it is more likely that X caused Y, or vice versa, given only data over their joint distribution. In this paper we propose a general inference framework based on Kolmogorov complexity, as well as a practical and computable instantiation based on the Minimum Description Length (MDL) principle.

Simply put, we propose causal inference by compression. That is, we infer that X is a likely cause of Y if we can better compress the data by first encoding X, and then encoding Y given X, than in the other direction. To show this works in practice, we propose ORIGO, an efficient method for inferring the causal direction from binary data. ORIGO employs the lossless PACK compressor and searches for that set of decision trees that encodes the data most succinctly. Importantly, it works directly on the data and does not require assumptions about neither distributions nor the type of causal relations.

To evaluate ORIGO in practice, we provide extensive experiments on synthetic, benchmark, and real-world data, including three case studies. Altogether the experiments show that ORIGO reliably infers the correct causal direction on a wide range of settings.

Keywords Causal Inference · Kolmogorov Complexity · MDL · Decision Trees · Binary Data

Kailash Budhathoki (✉)
Max Planck Institute for Informatics and Saarland University
Saarland Informatics Campus, Saarbrücken, Germany
E-mail: kbudhath@mpi-inf.mpg.de

Jilles Vreeken
Max Planck Institute for Informatics and Saarland University
Saarland Informatics Campus, Saarbrücken, Germany
E-mail: jilles@mpi-inf.mpg.de

1 Introduction

Causal inference, telling cause from effect, is perhaps one of the most important problems in science. To make absolute statements about cause and effect, carefully designed experiments are necessary, in which we consider representative populations, instrument the cause, and control for everything else [25]. In practice, setting up such an experiment is often very expensive, or simply impossible. The study of the effect of combinations of drugs is good example.

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, and 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. This even when we ignore the ethical aspect of potentially exposing volunteers to lethal side effects, as we need sufficiently many volunteers per combination of drugs, and all of these need to be (as) identical (as reasonably possible) for all other aspects, except the combination of drugs they get. That is, to investigate the combined effects of only 10 drugs, we already need $2^{10} = 1024$ groups, each of say 100 volunteers, meaning we would need to recruit over 100 000 near-identical volunteers. Clearly, this is not practically feasible.

We hence consider causal inference from *observational* data. That is, our goal is to infer the most likely direction of causation from data that has *not* been obtained in a completely controlled manner but is simply available. In recent years large strides have been made in the theory and practice of discovering causal structure from such data [12; 16; 25]. Most methods, and especially those that defined for pairs of variables, however, can only consider continuous-valued or discrete numeric data [27; 39] and are hence not applicable on binary data such as one would have in the above example.

We propose a general framework for causal inference on observational data, and give a practical instantiation for binary data. We base our inference framework on the solid foundations of Kolmogorov complexity [17; 20], and develop a score for pairs of data objects that identifies not only the direction [12], but also quantifies the *strength* of causation, without making any assumptions on the distribution nor the type of causal relation between the data objects, and without requiring any parameters to be set.

Kolmogorov complexity is not computable, however, and to be able to put it to practice we derive a practical, computable version based on the Minimum Description Length (MDL) principle [28; 9]. As a proof of concept, we propose ORIGO,[1] which is an efficient and parameter-free method for causal inference on binary data. ORIGO builds on the MDL-based PACK algorithm [36], and compresses data using decision trees. Simply put, it encodes the data one

---

[1] ORIGO is Latin for origin

attribute at a time using a decision tree. Such a tree may only split on previously encoded attributes. We use this mechanism to measure how much better we can compress the data of Y given the data of X, simply by (dis)allowing the trees for Y to split on attributes of 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 performs well in practice. It is robust to noise, dimensionality, and skew between cardinality of X and Y. It has high statistical power, and outperforms a recent proposal for discrete data by a wide margin. After discretization, ORIGO performs well on both univariate and multivariate benchmark data. Three case studies confirm that ORIGO provides intuitive results.

The main contributions of our work are as follows

- a theoretical framework for causal inference from observational data based on Kolmogorov complexity,
- a practical framework for causal inference based on MDL,
- a causal inference method for binary data, ORIGO.
- an extensive set of experiments on synthetic and real data.

This paper builds upon and extends Budhathoki and Vreeken [2]. In particular, we give a much more thorough introduction to causal inference by algorithmic information theory. We present our instantiation for binary data using decision trees in detail and self-contained, including the rationale of why decision tree models make sense, the exact encoding that we use, as well as show that it is an information score that can indeed be used for causal inference, and the algorithm for how to infer good models directly from data. Last, but not least, we provide a much extended set of empirical evaluations.

The remainder of this paper is organized as follows. We introduce notation and preliminaries in Section 2. Section 3 explains how to do causal inference based on Algorithmic Information Theory. In Section 4 we show how to derive practical, computable, causal indicators using the Minimum Description Length principle. We instantiate this framework for binary data using a decision-tree based compressor in Section 5. Related work is covered in Section 6, and we evaluate empirically in Section 7. We round up with discussion and conclusions in Sections 8 and 9, respectively.

All code and data are available for research purposes.[2]

2 Preliminaries

In this section, we introduce notations and background definitions we will use in subsequent sections.

---

[2] http://eda.mmci.uni-saarland.de/origo/

2.1 Notation

In this work, we consider *binary* data. We denote a binary string of length n by $s \in \{0,1\}^n$. A binary dataset D is a binary matrix of size n-by-m consisting of n rows, or transactions, and m columns, random variables, or attributes. A row is a binary vector of size m. We write $\Pr(X = v)$ for the probability of a random variable X assuming value v from the domain $dom(X)$. We say $X \rightarrow Y$ to indicate that X causes Y. We will model our data with sets of binary decision trees. The decision tree for X_i is denoted by T_i.

All logarithms are to base 2, and by convention we use $0 \log 0 = 0$.

2.2 Kolmogorov Complexity

To develop our causal inference principle, we need the concept of Kolmogorov complexity [17; 33; 3]. Below we give a brief introduction.

The Kolmogorov complexity of a finite binary string x, denoted $K(x)$, is the length of the shortest binary program p^* to a Universal Turing machine $\mathcal{U}$ that generates x and *halts*. Let $\ell(.)$ be a function that maps a binary string to its length, i.e. $\ell : \{0,1\}^* \rightarrow \mathbb{N}$. Then, $K(x) = \ell(p^*)$. More formally, the Kolmogorov complexity of a string x is given by

$$K(x) = \min\{\ell(p) \mid p \in \{0,1\}^* \text{ and } \mathcal{U}(p) = x\} \,,$$

where $\mathcal{U}(p) = x$ indicates that when the binary program p is run on $\mathcal{U}$, it generates x and halts. Intuitively, p^* is the most succinct *algorithmic* description of x, whereas $K(x)$ is then the length of the *ultimate* lossless compression of x.

Conditional Kolmogorov complexity, denoted $K(x \mid y)$, is the length of the shortest binary program p^* that generates x and halts when y is provided as an input to the program. We have $K(x) = K(x \mid \epsilon)$ where ϵ is the empty string.

Although Kolmogorov complexity is defined over binary strings, we can interchangeably use it over mathematical objects, or data objects in general, as any finite object can be encoded into a string. A data object can be a random variable, sequence of events, a temporal graph, etc.

The amount of *algorithmic information* contained in y about x is $I(y : x) = K(y) - K(y \mid x^*)$, where x^* is the shortest binary program for x. Intuitively, it is the number of bits that can be saved in the description of y when the shortest description of x is already known. Algorithmic information is symmetric, i.e. $I(y : x) \stackrel{\pm}{=} I(x : y)$, where $\stackrel{\pm}{=}$ denotes equality up to an additive constant, and therefore also called *algorithmic mutual information* [20]. Two strings x and y are *algorithmically independent* if they have no algorithmic mutual information, i.e. $I(x : y) \stackrel{\pm}{=} 0$.

For our purpose, we also need the Kolmogorov complexity of a distribution. The Kolmogorov complexity of a probability distribution P, $K(P)$, is the length of the shortest program that outputs $P(x)$ to precision q on input $\langle x, q \rangle$ [10]. More formally, we have

$$K(P) = \min\left\{|p| : p \in \{0,1\}^*, |\mathcal{U}(\langle x, \langle q, p \rangle \rangle) - P(x)| \leq 1/q\right\} \,.$$

We refer the interested reader to Li and Vitányi [20] for many more details on Kolmogorov complexity.

3 Causal Inference by Kolmogorov Complexity

Suppose we are given data over the joint distribution of two random variables X and Y of which we know they are dependent. We are interested in inferring the most likely causal relationship between X and Y. In other words, we want to infer whether X causes Y, whether Y causes X, or whether the two are merely correlated. To do so, we assume *causal sufficiency*. That is, we assume that there is no confounding variable Z that is the common cause of both X and Y.

We base our causal inference method on the following postulate.

Postulate 1 (Independence of Input and Mechanism [30]) *If X is the cause of Y, $X \rightarrow Y$, the marginal distribution of the cause $P(X)$, and the conditional distribution of the effect given the cause, $P(Y \mid X)$ are "independent"— $P(X)$ contains no information about $P(Y \mid X)$ and vice versa.*

We can think of conditional $P(Y \mid X)$ as the *mechanism* that transforms observations of X into observations of Y, i.e. generates effect Y for cause X. The postulate is plausible if this mechanism does not care how its input was generated, i.e. it is independent of $P(X)$. Importantly, this independence does *not* hold in the opposite direction as $P(Y)$ and $P(X \mid Y)$ both inherit properties from $P(Y \mid X)$ and $P(X)$ and hence will contain information about each other. This creates an asymmetry between cause and effect.

It is insightful to consider the example of solar power, where it is intuitively clear that the amount of radiation per cm^2 solar cell (cause) causes the generation of electricity in the cell (effect). It is relatively easy to change $P(cause)$ without affecting $P(effect \mid cause)$, as we can take actions such as, for example, moving the solar cell to a more sunny or more shady place, and varying its angle to the sun – note that while this will of course change the overall power output of the cell, it does not change the *conditional* distribution of the effect given the cause. If the same amount of radiation hits the cell, it will generate the same amount of power, after all. Likewise, it is easy to change $P(effect \mid cause)$ without affecting $P(cause)$. We can do so, for instance, by using more efficient cells – while this may again change the overall power output of the cell, it does not affect the distribution of the incoming radiation. It is surprisingly hard, however, to do the same in the anti-causal direction. That is, it is difficult to find actions that only change the distribution of the *effect*, $P(effect)$, while not affecting $P(cause \mid effect)$ or vice versa, as through their causal connection these two are intrinsically (more) dependent on each other.

The notion of *independence* in Postulate 1 is abstract, however. That is, to put the postulate to practice, one needs to choose and formalize an independence score. To this end, different formalisations have been proposed. Janzing et al. [16], for example, define independence in terms of information

geometry, Liu & Chan [21] formulate independence in terms of the distance correlation between marginal and conditional empirical distribution, whereas Janzing & Schölkopf [12] formalise independence using algorithmic information theory, and postulate *algorithmic* independence of $P(X)$ and $P(Y \mid X)$.

Since any physical process can be simulated on a Turing machine [7], it can, in theory, capture all possible dependencies that can be explained with a physical process. As such, the algorithmic model of causality has particularly strong theoretical foundations, and provides a better mathematical formalisation of Postulate 1. Using algorithmic independence, we arrive at the following postulate.

Postulate 2 (Algorithmic Independence of Markov kernels [12]) *If X is the cause of Y, $X \to Y$, the marginal distribution of the cause $P(X)$ and the conditional distribution of the effect given the cause $P(Y \mid X)$ are algorithmically independent, i.e. $I(P(X) : P(Y \mid X)) \overset{+}{=} 0$.*

The algorithmic independence between $P(X)$ and $P(Y \mid X)$ implies that the shortest description, in terms of Kolmogorov complexity, of the joint distribution $P(X, Y)$ is given by separate descriptions of $P(X)$ and $P(Y \mid X)$ [12]. As a consequence of the algorithmic independence of input and mechanism we have the following Theorem.

Theorem 1 (Simplest Factorization of the Joint Distribution [22]) *If X is the cause of Y, $X \to Y$,*

$$K(P(X)) + K(P(Y \mid X)) \le K(P(Y)) + K(P(X \mid Y)),$$

holds up to an additive constant.

That is, if X causes Y, factorising the joint distribution $P(X, Y)$ into $P(X)$ and $P(Y \mid X)$ will lead, in terms of Kolmogorov complexity, to simpler descriptions of the distributions than factorising it into $P(Y)$ and $P(X \mid Y)$. Note that the total complexity of the causal model $X \to Y$ is given by the complexity of the marginal distribution of the cause $P(X)$ and the complexity of the conditional distribution of the effect given the cause $P(Y \mid X)$.

With that, we can perform causal inference by simply identifying that direction between X and Y where factorization of the joint distribution yields the lowest total Kolmogorov complexity. Although this inference rule has sound theoretical foundations, Kolmogorov complexity is not computable — amongst which due to the *halting problem*. We can approximate Kolmogorov complexity from above, however, through lossless compression [20]. More generally, the Minimum Description Length (MDL) principle [28; 9] provides a statistically sound and computable means for approximating Kolmogorov complexity [37; 9]. Next, we discuss how MDL can be used for causal inference.

4 Causal Inference by Compression

The Minimum Description Length (MDL) [28] principle is a practical version of the Kolmogorov complexity. Both embrace the slogan *Induction by Compression*.

Instead of all possible programs, MDL considers only those programs for which we know they generate x and halt. That is, lossless compressors. The more powerful the compressor, the closer we are to Kolmogorov complexity. Ideal MDL, which considers all programs that generate x and halt, coincides with Kolmogorov complexity.

The MDL principle has its root in the two-part decomposition of Kolmogorov complexity [20, Ch. 5]. It can be roughly described as follows.

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

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

where

- *$L(M)$ is the length, in bits, of the description of the model, and*
- *$L(D \mid M)$ is the length, in bits, of the description of the data 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 $M \in \mathcal{M}$.

The algorithmic 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. 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 $K(P(\cdot))$.

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

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

where the first term is the total description length of X and M_X, and the second the total description length of Y and $M_{Y|X}$ given the data of X. We define $L_{Y \to X}$ analogously.

From Theorem 1, using the above indicators, we arrive at the following causal inference rules:

- If $L_{X \to Y} < L_{Y \to X}$, we infer $X \to Y$.
- If $L_{X \to Y} > L_{Y \to X}$, we infer $Y \to X$.
- If $L_{X \to Y} = L_{Y \to X}$, we are undecided.

That is, if total description length from X towards Y is simpler than vice versa, we infer X is likely the cause of Y under the causal mechanism represented by the used model class. If it is the other way around, we infer Y is likely the cause of X. The larger the difference between the two indicators, i.e. $|L_{X \to Y} - L_{Y \to X}|$, the stronger the causal explanation in one direction. If the total description length is the same in both directions, we are undecided. In practice, one can naturally introduce a threshold ϵ and treat differences between the two indicators smaller than ϵ as undecided.

To use these indicators in practice, we have to define what causal model class $\mathcal{M}$ we use, how to describe a model $M \in \mathcal{M}$ in bits, how to encode a dataset D given a model M, and how to efficiently approximate the optimal $M^* \in \mathcal{M}$. This we discuss in the next section.

5 Causal Inference by Tree-based Compressors

To apply the MDL-based causal inference rule in practice, we need a class of models suited for causal inference. As such, the model class must allow to causally explain Y given X and vice versa. One such model class is that of decision trees. A decision tree allows us to model dependencies on other attributes by splitting, i.e. conditionally describe the data of an attribute X_i given an attribute X_j. In other words, decision trees can model local dependencies between variables that can identify parts of the data that causally depend on each other. Note that this comes close to the spirit of average treatment effect in randomized experiments [29].

As models we consider sets of decision trees, such that we have one decision tree per attribute in the data. The dependencies between variables modelled by these trees induce a directed graph. To ensure lossless decoding, there needs to an order on the variables in a graph. It is easy to see that there exists an order of the variables if an only if the graph is *acyclic*. Hence, we enforce that there are no cyclic dependencies between variables across these trees.

In Figure 1, we give a toy example to show the valid models. For M_X and M_Y, we only allow dependencies between variables in X, and between variables in Y respectively, but not in between. In $M_{Y|X}$, we only allow variables in Y to acyclically depend on each other, as well as on variables in X. Therefore, for the causal model $M_{X \to Y}$, we allow variables in X to depend on each other, and variables in Y to depend on either X or Y. The reverse model $M_{Y \to X}$ is constructed analogously.

Next we instantiate the MDL-based causal inference framework for binary data. As such, we require a compressor for binary data that uses a set of decision trees as its model class. Importantly, the compressor should consider both the complexity of the model and that of the data under the model into account. One such compressor that fits our requirements is PACK [36]. In particular, we build upon PACK to instantiate the MDL-based causal score. Next we briefly explain how PACK works.
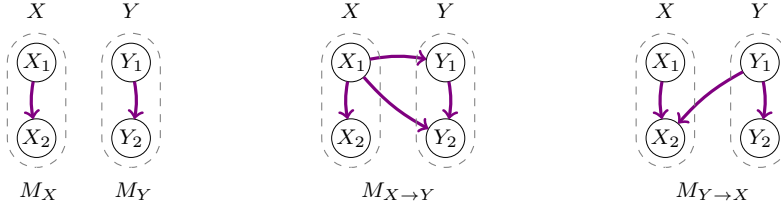
Fig. 1: A toy example of valid models. A directed edge from a node P to a node Q indicates that Q depends on P.

(a) Tree for X_1 (b) Tree for X_2 (c) Tree for X_3 (d) Dependency DAG
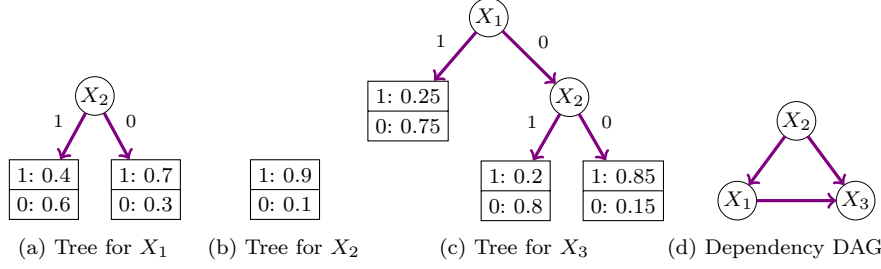
Fig. 2: In (a), (b), and (c), we give the example decision trees generated by PACK for a toy binary dataset containing three attributes, namely X_1, X_2, and X_3. In (d), we show the dependency graph for these trees.

5.1 Tree-based Compressor for Binary Data

PACK is an MDL based algorithm for discover interesting itemsets from binary data [36]. To do so, it discovers a set of decision trees that together encode the data most succinctly. The authors of PACK show there is a connection between interesting itemsets and paths in these trees [36]. 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 hypothetical binary dataset with three attributes X_1, X_2, and X_3. PACK aims at discovering the set of trees such that we can encode the whole data in as few as possible bits. In Figure 2a – 2c we give an example of the trees PACK could discover. As the figure shows, X_1 depends on X_2, and X_3 depends on both X_1 and X_2. These trees identify both local causal dependencies, as well as the global causal DAG shown in Figure 2d.

Let D be a binary data having n rows over m attributes X. We encode an attribute X_i using its decision tree T_i. Let M be a model that consists of a set of decision trees for the attributes, $M = \{T_1, T_2, \ldots T_m\}$. To encode an attribute X_i using its decision tree T_i over the complete data D, we use an optimal prefix code. For a probability distribution P on some finite set $\mathcal{S}$, the length of an optimal prefix code for a symbol $s \in \mathcal{S}$ is given by $-\log P(s)$ [5]. In particular, we encode each leaf $l \in lvs(T_i)$ of the tree. Hence the total cost

of encoding X_i using T_i over the complete data D is given by

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

where $P(X_i = v \mid l)$ is the empirical probability of $X_i = v$ given that leaf l is chosen, and n_{vl} is the number of samples in leaf l taking value v [36].

To decode the attributes, we need to transmit the decision trees as well. To this end, first we transmit the leaves of the decision trees. We use refined MDL [9, chap 1] to compute the complexity of a leaf $l \in lvs(T_i)$ as

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

where r is the number of rows for which the leaf l is used [36]. It can be computed in linear time for the family of multinomial distributions [18].

Then we encode the number of nodes in the decision tree T_i. In doing so, we use one bit to indicate whether the node is a leaf or an intermediate node. If the node is an intermediate node, we use an extra $\log m$ bits to identify the split attribute [36]. Let $intr(T_i)$ be the set of all intermediate nodes of a decision tree T_i. Then the number of bits needed to describe a decision tree T_i is given by

$$L(T_i) = \sum_{N \in intr(T_i)} (1 + \log m) + \sum_{l \in lvs(T_i)} (1 + L(l)) \, .$$

Therefore the total number of bits needed to describe the decision tree T_i, and describe X_i over the complete data D using T_i is given by

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

Putting it together, the total number of bits needed to describe all the trees, one for each attribute, and the complete data D is given by

$$L(D, M) = \sum_{T_i \in M} L(X_i, T_i) \, .$$

To discover good models directly from data, Tatti & Vreeken propose the GREEDYPACK algorithm [36]. For self containment, we give the pseudo-code of the main algorithm as Algorithm 1. We start with a model consisting of only trivial trees—simple tree without splitting on any other attributes as shown in Figure 2b—per attribute (line 1). To ensure that the decision tree model is valid, we build a dependency graph between attributes (line 2–3). We then proceed to iteratively discover the split that maximizes compression. To this end, for each attribute $X_i \in X$, we consider splitting on the other attributes X_j that we have not split on before, as long as the induced graph remains acyclic (line 5–9). We store the best split per attribute (line 10–12). Then we greedily select the overall best split, and iterate until no further split can be found that can save any bits (line 13–16). We refer the interested reader to the original paper [36] for more details on PACK.

Algorithm 1: GREEDYPACK

Input: A binary data D over m attributes X
Output: A set of binary decision trees $\{T_1, T_2, \ldots, T_m\}$

1 $T_i \leftarrow \text{TRIVIALTREE}(X_i)$ for $i = 1, 2, \ldots, m$;
2 $V \leftarrow \{1, 2, \ldots, m\}, E \leftarrow \phi$;
3 $G \leftarrow (V, E)$;
4 while $L(D, M)$ *decreases* **do**
5   **for** $X_i \in X$ **do**
6 $C_i \leftarrow T_i$;
7   **for** $l \in lvs(T_i)$ *and* $j = 1, 2 \ldots m$ **do**
8   **if** $E \cup (i, j)$ *is acyclic and* $j \notin path(l)$ **then**
9 $T \leftarrow \text{SPLITTREE}(T_i, l, X_j)$;
10   **if** $L(T) < L(C_i)$ **then**
11 $C_i \leftarrow T$;
12 $u_i \leftarrow j$;

13 $k \leftarrow \arg\min_i (L(C_i) - L(T_i))$;
14   **if** $L(C_k) < L(T_k)$ **then**
15 $T_k \leftarrow C_k$;
16 $E \leftarrow E \cup (k, u_k)$;

17 return $\{T_1, T_2, \ldots, T_m\}$

5.2 PACK as an Information Measure

The algorithmic independence of Markov kernels (Postulate 2) links observations to causality: we can reject a causal hypothesis if the algorithmic independence of Markov kernels is violated [12]. The notion of algorithmic independence, however, uses Kolmogorov complexity as an information measure, and is hence incomputable. While we know that MDL provides a well-founded way to approximate Kolmogorov complexity in general, the question remains whether this also holds for causal inference, and in particular, whether this holds for our PACK score. The answer is yes. Steudel et al [35] show that independence of Markov kernels is justified when we use a compressor as an information measure, if we restrict ourselves to the class of causal mechanisms that is adapted to the information measure. In general, let $\mathcal{X}$ be a set of discrete-valued random variables and Ω be the powerset of $\mathcal{X}$, i.e. the set of all subsets of $\mathcal{X}$. We then have the following definition of an information measure.

Definition 1 (Information measure [35]) A function $R : \Omega \to \mathbb{R}$ is an information measure if it satisfies the following axioms:

(a) *normalization*: $R(0) = 0$,
(b) *monotonicity*: $X \leq Y$ implies $R(X) \leq R(Y)$ for all $X, Y \in \Omega$,
(c) *submodularity*: $R(X \cup Z) - R(X) \geq R(Y \cup Z) - R(Y)$ for all $X, Y \in \Omega$, $X \subseteq Y$, and for all $Z \notin Y$.

This leaves us to show that PACK is an information measure, i.e. that it fulfills these properties. Let $L : \Omega \to \mathbb{R}$ be the PACK score.

(a) PACK trivially satisfies the normalization property.

(b) We examine the monotonicity property under subset restriction. If $X \subseteq Y$, we can decompose Y into X and Z such that $Y = X \cup Z$. Then $L(Y) = L(X \cup Z) = L(X) + L(Z \mid X) \geq L(X)$. This shows that PACK score is monotonic.

(c) We have $L(X \cup Z) - L(X) = L(Z \mid X)$ and $L(Y, Z) - L(Y) = L(Z \mid Y)$. Since $X \subseteq Y$, and providing PACK more possibilities to split on can only improve compression, $L(Z \mid X) \geq L(Z \mid Y)$. Therefore, $L(X \cup Z) - L(X) \geq L(Y \cup Z) - L(Y)$, which implies that PACK is submodular.

By which we have shown that PACK is indeed an information measure, and hence can pick up causal structure from observations where the causal mechanism is modelled by binary decision trees.

Next we discuss how to compute our MDL-based causal score using PACK.

5.3 Instantiating the MDL score with PACK

To compute $L(X, M_X)$, we can simply compress X using PACK. However, computing $L(Y, M_{Y|X} \mid X)$ is not straightforward, as PACK does not support conditional compression off-the-shelf. Clearly, it does not suffice to simply compress X and Y together as this gives us $L(XY, M_{XY})$ which may use any acyclic dependency between X and Y and vice versa. When computing $L_{X \to Y}$ or $L(Y, M_{Y|X})$, however, we do not want the attributes of X to depend on the attributes of Y. Therefore, we modify line 8 of GREEDYPACK such that an attribute of X is only allowed to split on other attributes of X, and an attribute of Y is allowed to split on both the attributes of X and the other attributes of 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. As the implementation of PACK already provides this feature, we do not have to binarise categorical data ourselves. Moreover, as we will see in the experiments, with a proper discretization, we can even reliably infer causal directions from discretized continuous real-valued data.

5.4 Computational Complexity

Next we analyse the computational complexity of ORIGO. To compute $L_{X \to Y}$, we have to run PACK only once. Greedy PACK uses the ID3 algorithm to construct binary decision trees, therewith the computational complexity of Greedy PACK is $\mathcal{O}(2^m n)$, where n is the number of rows in the data, and m the total number of attributes in X, and Y, i.e. $m = |X| + |Y|$. To infer the causal direction, we have to compute both $L_{X \to Y}$, and $L_{Y \to X}$. Therefore, in the worst case, the computational complexity of ORIGO is $\mathcal{O}(2^m n)$. In practice, ORIGO is fast, and completes within seconds.

6 Related Work

Inferring causal direction from observational data is a challenging task if no controlled ranodmized experiments are available. Due to its importance in practice, however, causal inference has recently seen increased attention [25; 34; 31; 12]. Most proposed causal inference frameworks are limited in practice, however, as they rely on strong assumptions, or have been defined only for either continuous real-valued, or discrete numeric data.

Constraint-based approaches like the conditional independence test [34; 25] require at least three observed random variables. Moreover, these constraint-based approaches cannot distinguish Markov equivalent causal DAGs [38] as the factorization of the joint distribution $P(X, Y)$ is the same in both direction, i.e. $P(X)P(Y \mid X) = P(Y)P(X \mid Y)$. Hence they cannot decide between $X \to Y$ and $Y \to X$.

There do exist methods that can infer the causal direction from two random variables. Generally, they exploit the sophisticated properties of the joint distribution. The linear trace method [14; 42] 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 [4] can infer non-linear causal relations, but requires the causal relation to be deterministic, functional, and invertible. In theory, we do not make any assumptions on the causal relation between variables.

One of the key frameworks for causal inference are the Additive Noise Models (ANMs) [31; 11; 41; 27]. The ANMs assume that the effect is governed by the cause and an additive noise, and the causal inference is done by finding the direction that admits such a model. Peters et al. [26] propose an ANM for discrete numeric data. However regression is not ideal for modelling nominal variables. Furthermore, it only works with univariate cause-effect pairs.

Algorithmic information theory provides a sound general theoretical foundation for causal inference [12]. As such, causality is defined in terms of the algorithmic similarity between data objects. In particular, for two random variables X and Y, if X causes Y, the shortest description of the joint distribution $P(X, Y)$ is given by the separate description of the marginal distribution of the cause $P(X)$ and the conditional distribution of the effect given the cause $P(Y \mid X)$ [12]. The algorithmic information theoretic viewpoint of causality is more general in the sense that any physical process can be simulated by a Turing Machine. Janzing & Steudel [13] use it to justify the ANM-based causal discovery.

Kolmogorov complexity, however, is not computable. To perform causal inference based on algorithmic information theoretic frameworks therefore requires (efficiently) computable notions of independence or information. The information-geometric approach [16] defines independence in terms of the orthogonality in information space. Sgouritsa et al [30] define independence in terms of the accuracy of the estimation of conditional distribution using corresponding marginal distribution. Janzing and Schölkopf [12] sketch how

comparing marginal distributions, and resource bounded computation could be used to infer causal direction, but do not give practical instantiations. Vreeken [39] proposed Ergo, 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. To apply this method in practice for univariate and multivariate continuous real-valued data, Vreeken instantiates it using cumulative entropy.

All above methods consider numeric data only. Causal inference on observational binary data has seen much less attention. The classic proposal by Silverstein et al [32] uses conditional independence test, and hence requires an independent variable Z to tell whether X and Y have any causal relation. A very recent proposal by Liu and Chan [21] defines independence in terms of the distance correlation between empirical distributions $P(X)$ and $P(Y \mid X)$, and propose Dc to infer the causal direction from nominal data. In the experiments, we will compare to Dc directly. In addition, we will compare to the Ergo score [39], instantiating it with Pack as $L(Y, M_{Y|X} \mid X)/L(Y, M_Y)$ and vice versa.

7 Experiments

We implemented Origo in Python and provide the source code for research purposes, 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 consider synthetic, benchmark, and real-world data. We compare Origo against the Ergo score [39] instantiated with Pack, and Dc [21].

7.1 Synthetic Data

To evaluate Origo on the data with known ground truth, we consider synthetic data. In particular, we generate binary data X and Y such that attributes in Y probabilistically depend on the attributes of X, termed here onwards as *dependency*. Throughout the experiments on synthetic data, we generate X of size 5000-by-k, and Y of size 5000-by-l.

To this end, we generate data on a per attribute basis. First, we assume the ordering of attributes – the ordering of attributes in X followed by the ordering of attributes in Y. Then, for each attribute, we generate a binary decision tree. In doing so, we only consider the attributes preceding it in the ordering as candidate nodes for its decision tree. Then, each row is generated by following the ordering of attributes, 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 attributes.

---

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

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. $X \rightarrow Y$. 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*), the percentage of indecisive inferences, and the percentage of incorrect inferences. We start with $k = l = 3$. We fix the split probability to 1.0, and generate trees with the maximum possible height, i.e. $k + l - 1 = 5$. In Figure 3a, 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 $k = l = 3$, and the split probability to 1.0. In Figure 3b, 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 $k = l = 3$, fix the dependency to 1.0, and generate trees with the maximum possible height. In Figure 4a, 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 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* [1]. 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 4b, we compare the accuracy of Origo against bagging (OrigoB) for symmetric cause effect pairs. We see that bagging does not really improve the result. This is not unexpected
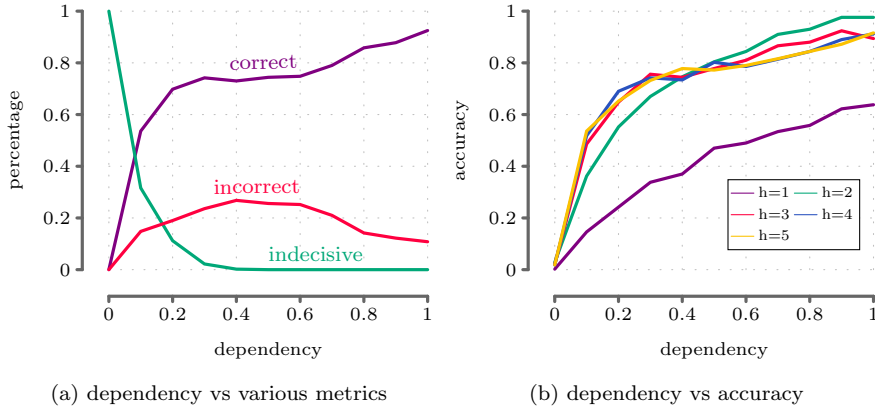
(a) dependency vs various metrics   (b) dependency vs accuracy

Fig. 3: For synthetic datasets with $k = l = 3$, we report (a) fraction of correct, incorrect and indecisive decisions at various dependencies (b) the accuracy at various dependencies for trees with various maximum heights.

as bagging is mainly a way to overcome overfitting, which by MDL we are naturally protected against [9]. These results confirm this conviction.

Next we investigate the accuracy of ORIGO on cause-effect pairs with asymmetric number of attributes. 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 $k = 1$, $l = 3$ and remaining 250 with $k = 3$, $l = 1$. In particular, we consider those pairs for correctness where there is at least one causal edge from X to Y. In Figure 5a, 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 $k = l = 3$. 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 X to Y. In Figure 5b, 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.

Dimensionality

Next we study the robustness against dimensionality. First we consider cause-effect pairs with symmetric number of attributes, i.e. $k = l$ 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 6a, we see that ORIGO is highly accurate in every setting.
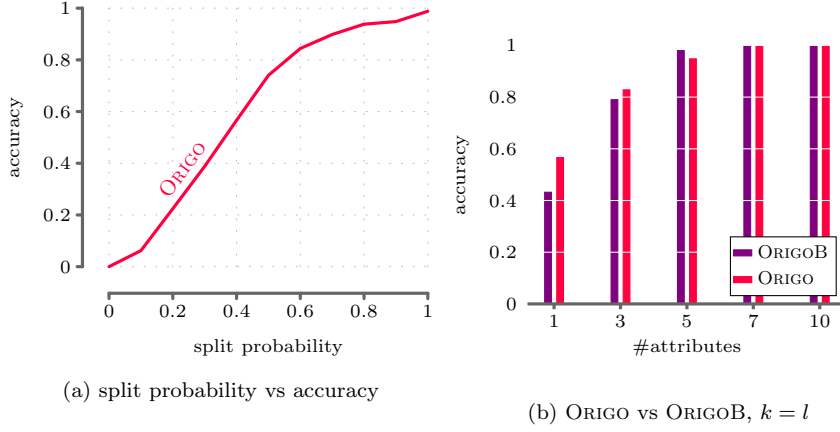
(a) split probability vs accuracy

(b) ORIGO vs ORIGOB, $k = l$

Fig. 4: For synthetic datasets, we show (a) the accuracy at various split probabilities for ORIGO with $k = l = 3$, and (b) compare the accuracy against bagging in symmetric case with $k = l$.

(a) dependency vs accuracy (1 vs 3)
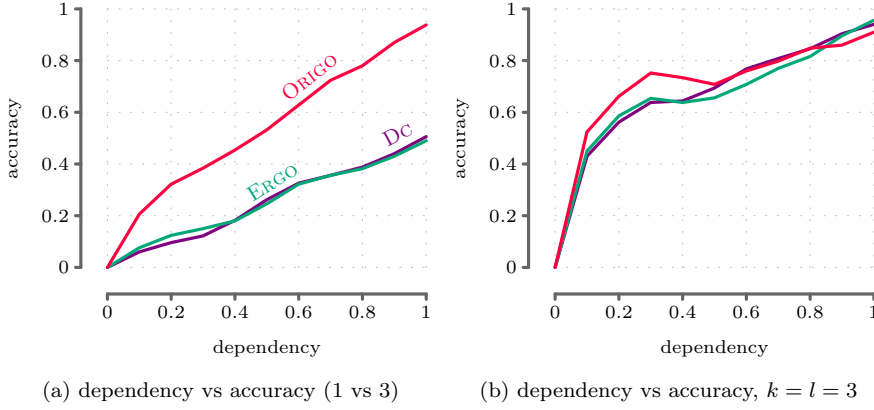
(b) dependency vs accuracy, $k = l = 3$

Fig. 5: For synthetic datasets, we compare (a) the accuracy in asymmetric case (1 vs. 3), and (b) the accuracy at various dependencies in symmetric case ($k = l = 3$)

With the exception of the univariate case, ERGO also performs well when both X and Y have the same cardinality.

In practice, however, we also encounter cause-effect pairs with asymmetric cardinalities. To evaluate performance in this setting, we set respectively k and l to 5 and vary the other between 1 to 10 – and generate 100 data pairs per setting. We see that ORIGO outperforms ERGO by a huge margin the stronger the unbalance between the cardinalities of X and Y. This is due to the inherent bias of ERGO favouring the causal direction from the side with
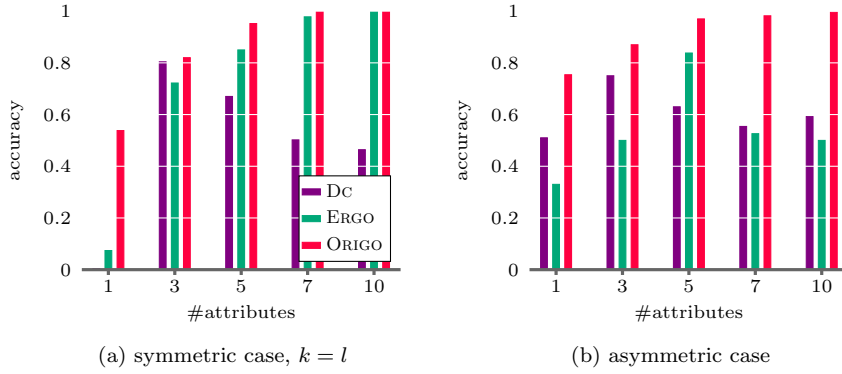
(a) symmetric case, $k = l$ (b) asymmetric case

Fig. 6: For synthetic datasets, we report the accuracy (a) in symmetric case with $k = l$, and (b) in asymmetric case (5 vs. varying cardinalities).

higher complexity towards the simple one. In addition, we see that Origo outperforms Dc in every setting.

Type I Error

To evaluate whether Origo infers relevant causal direction, we employ swap randomization [8]. Swap randomization is an approach for producing random datasets by altering the internal structure of the data while preserving its row and column margins. The internal structure of the data is altered by successive swap operations, which correspond to steps in a Markov chain process.

More formally, given a binary data matrix, D, with n rows and m columns, we randomly identify four cells in D characterised by combination of row indices $r_1, r_2 \in \{1, 2, \ldots, n\}$ and column indices $c_1, c_2 \in \{1, 2, \ldots, m\}$ such that $D_{r_1,c_1} \neq D_{r_1,c_2}$ and $D_{r_2,c_1} \neq D_{r_2,c_2}$ but $D_{r_2,c_1} = D_{r_1,c_2}$ and $D_{r_1,c_1} = D_{r_2,c_2}$. Then, we swap the values of these four cells either in clockwise or anti-clockwise direction. The swap operation is performed repeatedly until the data mixes sufficiently enough to break the internal structure of the data, also called mixing time of a Markov chain. Although there is no optimal theoretical bound for the mixing time of a Markov chain, Gionis et al. [8] empirically suggest the number of swap operations to be in the order of number of 1s in the data.

The key idea behind significance testing with swap randomization is to create several random datasets with the same row and column margins as the original data, run the data mining algorithm on those data, and see if the results differ significantly between the original data and random datasets.

Let $\Delta = |L_{X \to Y} - L_{Y \to X}|$. We compare the Δ value of the actual cause-effect pair to those of 500 swap randomized versions of the pair. We set $k = l = 3$, fix the dependency to 1.0, the probability of split to 1.0, and generate trees with the maximum possible height. The null hypothesis is that the Δ value of the actual data is likely to occur in random data. In Figure 7a, we show the histogram of the Δ values for 500 swap randomized pairs. The Δ value
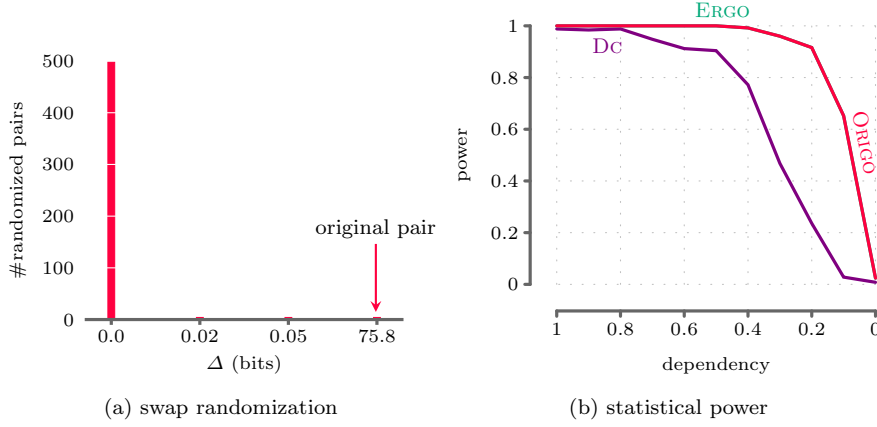
(a) swap randomization

(b) statistical power

Fig. 7: For synthetic datasets with $k = l = 3$, we show (a) the histogram of $\Delta = |L_{X \to Y} - L_{Y \to X}|$ values of 500 swap randomized cause-effect pairs using ORIGO, and (b) the statistical power at various dependencies.

of the actual cause-effect pair is indicated by an arrow. We observe that the probability of getting the Δ value of the actual data in a random data is zero, i.e. p-value = 0. Therefore, we can reject the null hypothesis at a much lower significance level.

Type II Error

To assess whether ORIGO identifies causal relationship when causal relationship really exists, we test its statistical power. The null hypothesis is that there is no causal relationship between cause-effect pairs. To determine the cut-off for testing the null hypothesis, we first generate 250 cause-effect pairs with no causal relationship. Then we compute their Δ values and set the cut-off Δ value at a significance level of 0.05. Next we generate new 250 cause-effect pairs with causal relationship. The statistical power is the proportion of the 250 new cause-effect pairs whose Δ value exceeds the cut-off delta value.

We set $k = l = 3$, the split probability to 1.0, and generate trees with the maximum possible height. We give the results in Figure 7b. The lines corresponding to ORIGO and ERGO overlap as both have the same high statistical power, outperforming DC in every setting.

Last, but not least, we observe that for all the above experiments inferring the causal direction for one pair typically takes only up to a few seconds. Next we evaluate ORIGO on real-world data.

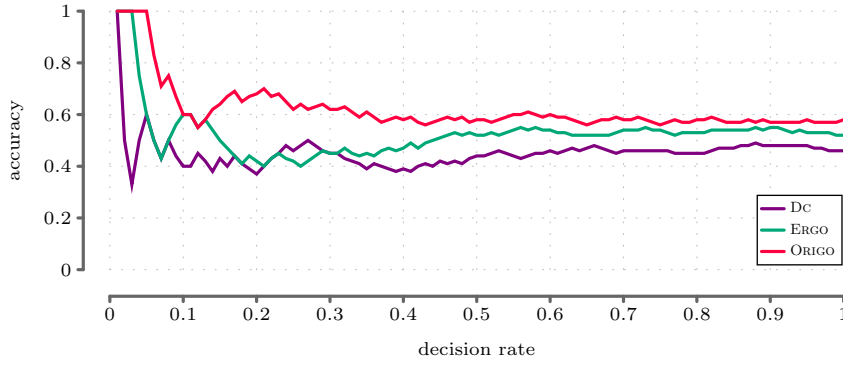7.2 Real-world Data

Next, we evaluate ORIGO on real-world data.

Fig. 8: Accuracy vs decision rate for univariate Tübingen cause-effect pairs discretized using IPD.

Univariate Pairs

First we evaluate ORIGO on benchmark cause-effect pairs with known ground truth [23]. In particular, we here consider the 95 univariate pairs. So far there does not exist a discretization strategy that provably preserves the causal relationship between variables. To complicate matters further we do not know the underlying domain of the data, and each cause-effect pair is from a different domain. Hence, for exposition we enforce one discretization strategy over all the pairs.

We considered various discretization strategies – including Equi-Frequency, and Equi-Width binning, MDL-based histogram density estimation [19], and parameter-free unsupervised interaction preserving discretization (IPD) [24]. Overall, we obtained the best results using IPD using its default parameters, and will report these below.

Next we investigate the accuracy of ORIGO against the fraction of decisions ORIGO is forced to make. To this end, we sort the pairs by their absolute score difference Δ in two directions in descending order. Then we compute the accuracy over top-k% pairs. The decision rate is the fraction of *top* cause-effect pairs that we consider. Alternatively, it is also the fraction of cause-effect pairs whose Δ is greater than some threshold Δ_t. For undecided pairs, we flip a coin. For other methods, we follow the similar procedure with their respective absolute score difference.

In Figure 8, we show the accuracy versus the decision rate for the benchmark univariate cause-effect pairs. If we look over all the pairs, we find that ORIGO infers correct direction in roughly 58% of all pairs. When we consider only those pairs where Δ is relatively high, i.e. those pairs where ORIGO is most decisive we see that over the top 8% most decisive pairs it is 75% accurate, yet still 70% accurate for the top 21% pairs, which is comparable with the top-performing causal inference frameworks for continuous real-valued data [16; 27; 30].

Table 1: Results on Tübingen multivariate cause-effect pairs [23]. "✓" means the correct causal direction is inferred, "×" means the wrong direction, and "−" means indecision.

Dataset	#rows	$|X|$	$|Y|$	Truth	ORIGO	ERGO	DC
Weather forecast	10 226	4	4	$Y \to X$	−	✓	−
Ozone	989	1	3	$Y \to X$	✓	✓	×
Auto-Mpg	392	3	2	$X \to Y$	✓	✓	×
Radiation	72	16	16	$Y \to X$	×	×	×
Chemnitz	1 440	3	7	$X \to Y$	✓	×	✓
Car	1 728	6	1	$X \to Y$	✓	✓	✓

Multivariate Pairs

Next we evaluate ORIGO quantitatively on real-world data with multivariate pairs. For that we consider four cause-effect pairs with known ground truth taken from [23]. The *Chemnitz* dataset is taken from Janzing et al. [15], whereas the *Car* dataset is from the UCI repository.[4] We again use IPD to discretize the data. We give the base statistics in Table 1. For each pairs, we report the number of rows, the number of attributes in X, the number of attributes in Y, the ground truth. Furthermore, we report the results of ORIGO, ERGO, and DC.

We find 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.

7.3 Qualitative Results

Last, we consider whether ORIGO provides results that agree with intuition. To this end we consider three case studies.

Acute inflammation

The *Acute inflammation* data is taken from the UCI repository.[4] It consists of the presumptive 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).

---

[4] https://archive.ics.uci.edu/ml/

Table 2: Results of ORIGO on *ICDM*. We give 8 characteristic and non-redundant exemplars drawn from top 17 causal directions.

discovered causal direction	Δ (bits)
frequent itemset $\rightarrow$ mining	4.809964
fp $\rightarrow$ tree	0.880654
drift $\rightarrow$ concept	0.869090
anomaly $\rightarrow$ detection	0.804479
lda $\rightarrow$ linear	0.772805
neural $\rightarrow$ network	0.748579
walk $\rightarrow$ random	0.701649
social $\rightarrow$ network	0.694999

We discretize 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 X, Y where $X = \{X_{11}, X_{12}, X_3, X_4, X_5, X_6\}$ and $Y = \{Y_1, Y_2\}$. We find that $Y \rightarrow X$. That is, ORIGO infers that the diseases cause the symptoms, which is in agreement with intuition.

ICDM abstracts

Next we consider the *ICDM abstracts* dataset, which is available from the authors of [6]. This dataset consists of abstracts – stemmed and stop-words removed – of 859 papers published at the ICDM conference until the year 2007. Each abstract is represented by a row and words are the attributes.

We use OPUS MINER on the ICDM abstracts dataset to discover top 100 self-sufficient itemsets [40]. Then, we apply ORIGO on those 100 self-sufficient itemsets. We sort the discovered causal directions by their Δ value in descending order. In Table 2, we give 8 highly characteristic and non-redundant results along with their Δ values taken from top 17 causal directions. We expect the causal directions having higher Δ values to show clear causal connection, and indeed we see that this is the case.

For instance, frequent itemset mining is one of the core topics in data mining. Clearly when *frequent itemset* appears in a text, it gives more information about the word *mining* than vice versa because mining could be about data mining, process mining, etc. among others. Likewise, *neural* gives more information about the word *network* than the other way around. Overall, the causal directions discovered by ORIGO in the ICDM dataset are sensible.

Census

The *Adult* dataset is taken from the UCI repository and consists of $48\,832$ records from the census database of the US in 1994. Out of 14 attributes, we consider only four – *work-class*, *education*, *occupation*, and *income*. In particular, we binarise *work-class* attribute into four attributes as *private, self-employed,*

Table 3: Results of Origo on *Adult*. We give 5 characteristic and non-redundant exemplars drawn from top 7 causal directions.

discovered causal direction	Δ (bits)
public-servant admin hs-graduate $\rightarrow \leq 50K$	9.917098
public-servant professional doctorate $\rightarrow > 50K$	8.053542
bachelors self-employed white-collar $\rightarrow > 50K$	7.719200
public-servant professional masters $\rightarrow > 50K$	7.583210
hs-graduate blue-collar $\rightarrow \leq 50K$	5.209738

public-servant, and *unemployed*. We binarise *education* attribute into seven attributes as *dropout, associates, bachelors, doctorate, hs-graduate, masters*, and *prof-school*. Further, we binarise *occupation* attribute into eight attributes as *admin, armed-force, blue-collar, white-collar, service, sales, professional*, and *other-occupation*. Lastly, we binarise *income* attribute into two attributes as $> 50K$ and $\leq 50K$.

We run Opus Miner on the resulting data and get top 100 self-sufficient itemsets. Then we apply Origo on those 100 self-sufficient itemsets. In Table 3, we report 5 interesting and non-redundant causal directions identified by Origo drawn from the top 7 strongest causal directions. Inspecting the results, we see that Origo infers sensible causal directions from the *adult* dataset. For instance, a professional with a doctorate degree working in a public office causes them to earn more than 50K per annum. However, working in a public office in an administrative position with a high school degree causes them to earn less than 50K per annum.

These case studies show that Origo discovers sensible causal directions from real-world data.

8 Discussion

The experiments show that Origo works well in practice. Origo reliably identifies true causal structure regardless of cardinality, skew, with high statistical power, even at low level of causal dependencies. On benchmark data it performs very well, despite information loss through discretization. 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 which is likely to improve the results.

Further, we rely on discretization strategies to discretize continuous real-valued data. We observe different results on continuous real-valued data depending on the discretization strategy we pick. It would make an engaging future

work to devise a discretization strategy for continuous real-valued data that preserves causal dependencies. Alternatively, it will be interesting to instantiate the framework using regression trees to directly consider real-valued data. This is not trivial, as it requires both a encoding scheme for this model class, as well as efficient algorithms to infer good sets of trees.

Our framework is based on causal sufficiency assumption. Extending ORIGO to include confounders is another avenue of future work. Moreover, our inference principle is defined over data in general, yet we restricted our analysis to binary, categorical, and continuous real-valued data. It would be interesting to apply our inference principle on time-series data. To instantiate our MDL framework the only thing we need is a lossless compressor that can capture directed relations on multivariate time-series data.

9 Conclusion

We considered causal inference from observational data. We proposed a framework for causal inference based on Kolmogorov complexity, and gave a generally applicable and computable framework based on the Minimum Description Length (MDL) principle.

To apply the framework in practice, we proposed ORIGO, an efficient method for inferring the causal direction from binary data. ORIGO uses decision trees to encode data, works directly on the data and does not require assumptions about neither distributions nor the type of causal relations. Extensive evaluation on synthetic, benchmark, and real-world data showed that ORIGO discovers meaningful causal relations, and outperforms the state-of-the-art.

Acknowledgements

References

1. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
2. Budhathoki K, Vreeken J (2016) Causal inference by compression. In: Proceedings of the 16th IEEE International Conference on Data Mining (ICDM), Barcelona, Spain, IEEE
3. Chaitin GJ (1969) On the simplicity and speed of programs for computing infinite sets of natural numbers. Journal of the ACM 16(3):407–422
4. Chen Z, Zhang K, Chan L (2013) Nonlinear causal discovery for high dimensional data: A kernelized trace method. In: Proceedings of the 13th IEEE International Conference on Data Mining (ICDM), Dallas, TX, pp 1003–1008
5. Cover TM, Thomas JA (2006) Elements of Information Theory. Wiley-Interscience New York

6. De Bie T (2011) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Mining and Knowledge Discovery 23(3):407–446

7. Deutsch D (1985) Quantum theory, the church-turing principle and the universal quantum computer. Proceedings of the Royal Society A (Mathematical, Physical and Engineering Sciences) 400(1818):97–117

8. Gionis A, Mannila H, Mielikäinen T, Tsaparas P (2007) Assessing data mining results via swap randomization. ACM Transactions on Knowledge Discovery from Data 1(3):167–176

9. Grünwald P (2007) The Minimum Description Length Principle. MIT Press

10. Grünwald PD, Vitányi PMB (2008) Algorithmic information theory. CoRR abs/0809.2754

11. Hoyer P, Janzing D, Mooij J, Peters J, Schölkopf B (2009) Nonlinear causal discovery with additive noise models. In: Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS), pp 689–696

12. Janzing D, Schölkopf B (2010) Causal inference using the algorithmic markov condition. IEEE Transactions on Information Technology 56(10):5168–5194

13. Janzing D, Steudel B (2010) Justifying additive noise model-based causal discovery via algorithmic information theory. Open Systems and Information Dynamics 17(2):189–212

14. Janzing D, Hoyer P, Schölkopf B (2010) Telling cause from effect based on high-dimensional observations. In: Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, pp 479–486

15. Janzing D, Hoyer P, Schölkopf B (2010) Telling cause from effect based on high-dimensional observations. In: Proceedings of the 27th International Conference on Machine Learning, International Machine Learning Society, pp 479–486

16. Janzing D, Mooij J, Zhang K, Lemeire J, Zscheischler J, Daniušis P, Steudel B, Schölkopf B (2012) Information-geometric approach to inferring causal directions. Artificial Intelligence 182-183:1–31

17. Kolmogorov AN (1965) Three approaches to the quantitative definition of information. Problems of Information Transmission 1:1–7

18. Kontkanen P, Myllymäki P (2007) A linear-time algorithm for computing the multinomial stochastic complexity. Information Processing Letters 103(6):227–233

19. Kontkanen P, Myllymäki P (2007) MDL histogram density estimation. In: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS), San Juan, Puerto Rico

20. Li M, Vitányi P (1993) An Introduction to Kolmogorov Complexity and its Applications. Springer

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

22. Mooij JM, Stegle O, Janzing D, Zhang K, Schölkopf B (2010) Probabilistic latent variable models for distinguishing between cause and effect. In: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Curran, pp 1687–1695

23. Mooij JM, Peters J, Janzing D, Zscheischler J, Schölkopf B (2016) Distinguishing cause from effect using observational data: Methods and benchmarks. Journal of Machine Learning Research 17(32):1–102

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

25. Pearl J (2000) Causality: Models, Reasoning, and Inference. Cambridge University Press, New York, NY, USA

26. Peters J, Janzing D, Schölkopf B (2010) Identifying cause and effect on discrete data using additive noise models. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pp 597–604

27. Peters J, Mooij J, Janzing D, Schölkopf B (2014) Causal discovery with continuous additive noise models. Journal of Machine Learning Research 15:2009–2053

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

29. Rubin DB (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of Educational Psychology 66(5):688–701

30. Sgouritsa E, Janzing D, Hennig P, Schölkopf B (2015) Inference of cause and effect with unsupervised inverse regression. In: Proceedings of the International Conference on

Artificial Intelligence and Statistics (AISTATS), Journal of Machine Learning Research, pp 847–855

31. Shimizu S, Hoyer PO, Hyvärinen A, Kerminen A (2006) A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research 7:2003–2030

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

33. Solomonoff RJ (1964) A formal theory of inductive inference. part I, II. Information and Control 7:1–22224–254

34. Spirtes P, Glymour C, Scheines R (2000) Causation, Prediction, and Search. MIT press

35. Steudel B, Janzing D, Schölkopf B (2010) Causal markov condition for submodular information measures. In: Proceedings of the 23rd Annual Conference on Learning Theory, OmniPress, pp 464–476

36. Tatti N, Vreeken J (2008) Finding good itemsets by packing data. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy, pp 588–597

37. Vereshchagin N, Vitanyi P (2004) Kolmogorov's structure functions and model selection. IEEE Transactions on Information Technology 50(12):3265–3290

38. Verma T, Pearl J (1991) Equivalence and synthesis of causal models. In: Proceedings of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI), pp 255–270

39. Vreeken J (2015) Causal inference by direction of information. In: Proceedings of the SIAM International Conference on Data Mining (SDM), Vancouver, Canada, pp 909–917

40. Webb G (2011) Filtered-top-k association discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(3):183–192

41. Zhang K, Hyvärinen A (2009) On the identifiability of the post-nonlinear causal model. In: Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence (UAI), pp 647–655

42. Zscheischler J, Janzing D, Zhang K (2011) Testing whether linear equations are causal: A free probability theory approach. In: Proceedings of the 27nd International Conference on Uncertainty in Artificial Intelligence (UAI), AUAI Press, pp 839–847

Author Biographies

Kailash Budhathoki is a PhD student at the Max Planck Institute for Informatics and Saarland University. He holds a PhD Fellowship from the International Max Planck Research School for Computer Science (IMPRS-CS). His research interests include how to discover assocations, correlations, and causation from data by means of algorithmic information theory. At time of writing he has not won any noteworthy awards yet.

Jilles Vreeken leads the Exploratory Data Analysis group at the DFG Cluster of Excellence on Multimodal Computing and Interaction (MMCI) at Saarland University, Saarbrücken, Germany. In addition, he is a Senior Researcher at the Max Planck Institute for Informatics. His research interests include virtually all topics in data mining and machine learning. He authored over 70 conference and journal papers, 3 book chapters, won the 2010 ACM SIGKDD Doctoral Dissertation Runner-Up Award, and won two best (student) paper awards. He likes to travel, to think, and to think while travelling.