# Causal Inference by Compression

Kailash Budhathoki and Jilles Vreeken
Max Planck Institute for Informatics and Saarland University, Saarbrücken, Germany
{kbudhath,jilles}@mpi-inf.mpg.de

***Abstract*—Causal inference is one of the fundamental problems in science. In recent years, several methods have been proposed for discovering causal structure from observational data. These methods, however, focus specifically on numeric data, and are not applicable on discrete nominal or binary data.**

**In this work, we focus on causal inference for binary data. Simply put, we propose causal inference by compression. To this end we propose an inference framework based on solid information theoretic foundations, i.e. Kolmogorov complexity. However, Kolmogorov complexity is not computable, and hence we propose a practical and computable instantiation based on the Minimum Description Length (MDL) principle.**

**To apply the framework in practice, we propose ORIGO, an efficient method for inferring the causal direction from binary data. ORIGO employs the lossless PACK compressor, 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 shows that ORIGO discovers meaningful causal relations, and outperforms state-of-the-art methods by a wide margin.**

## I. 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 [1]. In practice, setting up these experiments is very expensive, or even impossible.

The study of the effect of combinations of drugs is good example. Combining drugs can be positive as the overall effect may be amplified, such as used in combination treatment of HIV and cancer, but can also be negative as there can be severe and possibly lethal side effects. Even without considering the ethical side, for all but the smallest number of drugs there are so many possible combinations that it becomes practically impossible to test all of these in a controlled manner.

We hence consider causal inference from observational data. That is, the 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 observational data [2], [3]. Most methods, and especially those that defined for pairs of variables, however, can only consider continuous-valued or discrete numeric data [4], [5] 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. We base it on the solid foundations of Kolmogorov complexity [6], [7], and develop a score for pairs of data objects that identifies not only the direction [2], but also quantifies the *strength* of causation, all the while being unbiased to the complexities of the individual objects, 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 hence we derive a practical, computable version based on the Minimum Description Length (MDL) principle [8], [9]. To infer causal directions from binary data we propose ORIGO.[1] ORIGO is both efficient and parameter-free. It builds on the MDL-based PACK algorithm [10], and compresses data using decision trees. Simply put, it encodes the data one 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.

Extensive experiments on synthetic, benchmark, and real-world data show that ORIGO performs very well in practice. It is highly robust to noise, dimensionality, and skew between cardinality of $X$ and $Y$. It has very high statistical power, and outperforms a recent proposal for discrete data by a wide margin. Further, ORIGO performs surprisingly well on continuous-valued benchmark data after discretisation. Moreover, the case studies confirm it 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,
- an unbiased indicator that quantifies the causal direction between pairs of data objects $X$ and $Y$,
- 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.

The remainder of this paper is organized as follows. We introduce notation and preliminaries in Section II. Section III explains how to do causal inference based on Algorithmic Information Theory. In Section IV we instantiate our rules for binary data using a decision-tree based compressor. Related work is covered in Section V, and we evaluate empirically in Section VI. We round up with discussion and conclusions in Sections VII and VIII, respectively.

[1] ORIGO is Latin for origin

## II. Preliminaries

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

### A. Notation

In this work, we consider *binary* data. A *binary* dataset $D$ is a binary matrix of size $n$-by-$m$ consisting of $n$ rows, or transactions, and $m$ columns, or attributes. A row is a binary vector of size $m$. We use the notation $\Pr(a_i = v)$ to express the probability of an attribute $a_i$ assuming a value $v$. The decision tree for $a_i$ is denoted by $T_i$. All logarithms are to base 2, and by convention we use $0 \log 0 = 0$.

### B. Kolmogorov Complexity

To develop our causal inference principle, we need the concept of Kolmogorov complexity [6], [11], [12]. Below we give a brief introduction.

The Kolmogorov complexity of a 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. $\ell : \{0,1\}^* \to \mathbb{N}$. Then, $K(x) = \ell(p^*)$. More formally, the Kolmogorov complexity of a string $x$ is given by

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

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

Although Kolmogorov complexity is defined over finite binary strings, we can interchangeably use it over data objects as any finite data object can be encoded into a string [7].

To derive our causal inference rule, we need the Kolmogorov complexity of a set of data objects. For a set of data objects $X$, its Kolmogorov complexity, denoted $K(X)$, is the length of the shortest binary program to the Universal Turing machine $\mathcal{U}$ computes the listing of the elements of $X$ and *halts* [13]. That is, if $X = \{x_1, x_2, \ldots, x_n\}$, then $K(X)$ is given by

$$K(X) = \min\{\ell(p) \mid p \in \{0,1\}^* \text{ and } \mathcal{U}(p) = \langle x_1, \langle x_2, \ldots, \langle x_{n-1}, x_n\rangle \ldots\rangle\rangle\} \quad ,$$

where $\mathcal{U}(p) = \langle x_1, \langle x_2, \ldots, \langle x_{n-1}, x_n\rangle \ldots\rangle\rangle$ indicates that when the binary program $p$ is run on $\mathcal{U}$, it generates the listing of the elements in $X$. In turn, the conditional complexity $K(X \mid Y)$ is the length of the shortest binary program to a Universal Turing machine that generates listing of elements of $X$ given set of data objects $Y$ as auxiliary information.

We refer the interested reader to Li & Vitányi [7] for more details on Kolmogorov complexity.

## III. Causal Inference by Algorithmic Information Theory

Suppose we are given two sets of data objects $X$ and $Y$ that are correlated. We are interested in inferring the causal relationship between $X$ and $Y$. In other words, we want to infer whether $X$ causes $Y$, whether $Y$ causes $X$, or they are only correlated. To do so, we assume causal sufficiency. That is, we assume there are no confounders – there are no hidden common causes $Z$ of $X$ and $Y$.

Loosely speaking, we infer that $X$ is likely a cause of $Y$ if the shortest joint description of $X$ and $Y$ is given by the description of $X$ followed by the description of $Y$ given $X$. Intuitively, we deduce that $X$ causes $Y$, if it is easier to describe $X$ first, and then to describe $Y$ given $X$ than the other way around. We use $X \to Y$ to indicate $X$ causes $Y$, and $Y \to X$ for the other way around.

### A. Causal Inference by Kolmogorov Complexity

Next, we develop our causal inference rule from the ground on up using Kolmogorov complexity.

A cornerstone postulate in causal inference states that if $X$ causes $Y$, it is easier to describe $Y$ using $X$ than the other way around [1]. From Algorithmic Information Theory (AIT) standpoint, this means, if $X$ causes $Y$, $X$ has more information about $Y$ than the other way around. Therefore, if we can measure the amount of information $X$ provide towards the most succinct algorithmic description of $Y$, and vice versa, we can perform causal inference.

In terms of AIT, if $X$ causes $Y$, the shortest program $P_{Y|X}$ that computes $Y$ from $X$, will be much simpler than the shortest program $P_{X|Y}$ that computes $X$ from $Y$. This algorithmic viewpoint of causality bears close resemblance to its statistical counterpart – the functional causal model [1] – where an effect is modelled as a function of its cause, and an unobserved noise term. Intuitively, program $P_{Y|X}$ consists of two parts; the compressible part of $Y$ given $X$, i.e. the causal mechanism, or function that generates $Y$ from $X$, and the incompressible part of $Y$ given $X$, which is the randomness specific to $Y$ and independent of $X$, which is necessary to reconstruct the exact observed values of $Y$ given $X$. Importantly, $P_{Y|X}$ will be much simpler than $P_{X|Y}$, the program that generates $X$ from $Y$, as it does not have to 'reverse-engineer' the process. Therefore, in terms of the Kolmogorov complexity, we expect $K(Y \mid X) < K(X \mid Y)$ if $X$ causes $Y$.

To infer the causal direction, we can take the absolute difference between $K(Y \mid X)$ and $K(X \mid Y)$. However, in practice, the complexities of $X$ and $Y$ are different. As a result, inferring causal direction based on the absolute difference would be biased towards the simplest object. Therefore, to reliably identify the correct direction, we have to normalise.

One way to normalise is to consider the *relative conditional complexity* [5]. That is, we look at the reduction in the complexity of $Y$ knowing $X$. More formally, the relative

amount of directed information from $X$ to $Y$ using relative conditional complexity, denoted $\delta_{X\to Y}$, is

$$\delta_{X\to Y} = \frac{K(Y \mid X)}{K(Y)} \quad , \tag{1}$$

and we define $\delta_{Y\to X}$ analogous. $\delta_{X\to Y}$ takes a value of 1 when $X$ has no information about $Y$ and will be close to 0 when $X$ has all the information about $Y$. If $\delta_{X\to Y} < \delta_{Y\to X}$, we infer that $X$ is likely to have caused $Y$ than vice versa. Alternatively, if $\delta_{Y\to X} < \delta_{X\to Y}$, we infer $Y \to X$.

While better than simple conditional complexity, the relative conditional complexity still suffers from undue bias, this time towards the more complex set of data objects. That is, when we have two sets of data objects with different complexities, the relative conditional complexity favours the causal direction from more complex set of data objects towards the more simple one.

More formally, let $X$ be more complex than $Y$, i.e. $K(X) > K(Y)$. Now even if both $X$ and $Y$ contribute equal amount of information in the succinct description of each other, we will have $\delta_{X\to Y} < \delta_{Y\to X}$ due to different quantities in the normalisation in both directions. To make things worse, relative conditional complexity is also prone to free-rider causal rules. To show this, by slightly abusing the notation, we use $xz$ for a set $\{x, z\}$ and $K(y \mid xz)$ for $K(y \mid x, z)$. By the property of Kolmogorov complexity, we have $K(y \mid xz) \leq K(y)$. As a result, $\delta_{xz\to y} \leq \delta_{x\to y}$. In particular, if $x$ is the cause of $y$ then we are likely to infer $xz \to y$. That is, other data objects can piggyback on $x$ thereby giving us redundant causal rules.

One way to cope with this problem is to consider the complexity of both $X$ and $Y$. That is, instead of using relative conditional complexity, we take the *relative joint complexity*. We define the relative amount of directed information from $X$ to $Y$ using relative joint complexity, denoted $\Delta_{X\to Y}$, as

$$\Delta_{X\to Y} = \frac{K(X) + K(Y \mid X)}{K(X) + K(Y)} \quad , \tag{2}$$

and again defining $\Delta_{Y\to X}$ analogously. Importantly, the normalising term is the same on both directions – there is no bias due to the complexities of the individual objects.

It is important to note that in general the symmetry of information implies $K(X) + K(Y \mid X) \stackrel{+}{=} K(Y) + K(X \mid Y)$ [7]. As a result, the literature often claim that cause cannot be distinguished from an effect using their joint description. However, a recent result by Janzing & Schölkopf [2] shows that the additive equality does *not* hold when $X$ causes $Y$ or vice versa. In other words, if $X \to Y$, there is indeed an asymmetry between cause and effect. This is exactly the asymmetry we use in our causal inference rule.

In particular, Equation 2 takes a value of 1 when $X$ has no information about $Y$. The smaller the value ($< 1$), the more the information $X$ has about $Y$. If $\Delta_{X\to Y} < \Delta_{Y\to X}$, $X$ has more information about $Y$ and by the direction of information, we infer that $X$ is likely to be the cause of $Y$ than vice versa. If $\Delta_{Y\to X} < \Delta_{X\to Y}$, we infer $Y \to X$. If $\Delta_{X\to Y} = \Delta_{Y\to X}$, we are undecided.

Causal inference using algorithmic information theory has a number of powerful properties. First, we do not need to assume the distribution of data as we only need to consider the data objects. Second, the inference rule is *generic* in the sense that we are not restricted to one type of data. Third, we do not need to assume any specific kind of causal mechanism between $X$ and $Y$, nor do we need to assume anything about the shape or type of noise within the data.

Although Kolmogorov complexity has sound theoretical foundations, due to the widely known *halting problem* it is not computable. We can approximate Kolmogorov complexity from above through lossless compression, however. More generally, the Minimum Description Length (MDL) principle provides a statistically sound and computable means for approximating Kolmogorov complexity. Next, we discuss how MDL can be used for causal inference.

### *B. Causal Inference by MDL*

The Minimum Description Length (MDL) [8] principle is a practical version of the Kolmogorov complexity. It circumvents the computability issue of the Kolmogorov complexity by restricting the programs to those that always *halt*, yet are general enough to allow us to capture most of the regularities. Moreover, we can select these programs using our prior knowledge of the problem domain. For instance, if we have data sets that might be seen as the points of a polynomial, we can choose programs or models based on the class of all polynomials.

The MDL principle has its root in the two-part decomposition of the Kolmogorov complexity [7]. It can be roughly described as follows [9]. Given a set of models $\mathcal{M}$ and data $D$, the best model $M \in \mathcal{M}$ is the one that minimises $L(M) + L(D \mid M)$, 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 the model $M$. Intuitively $L(M)$ represents the compressible part of the data, and $L(D \mid M)$ represents the noise in the data.

For our goal we will need a model class $\mathcal{M}$, containing causal models $M_{X\to Y} = (M_X, M_{Y|X})$ where $M_X$ is the model that describes the structure of $X$, and $M_{Y|X}$ is a model for $Y$ given $X$. Now assuming that we have already defined our causal model class $\mathcal{M}$, we can approximate $K(X)$ using MDL by $L(X, M_X)$, which is defined as

$$L(X, M_X) = L(M_X) + L(X \mid M_X) \quad ,$$

where $L(M_X)$ is the length, in bits, of the description of the MDL optimal model for $X$, and $L(X \mid M_X)$ is the length, in bits, of the description of $X$ when encoded with $M_X$. Likewise for $K(Y)$. We can approximate $K(Y \mid X)$ by $L(Y, M_{Y|X} \mid X)$, which is defined as

$$L(Y, M_{Y|X} \mid X) = L(M_{Y|X}) + L(Y \mid M_{Y|X}, X) \quad ,$$

where $L(M_{Y|X})$ is the length, in bits, of the description of the causal model from $X$ to $Y$, and $L(Y \mid M_{Y|X}, X)$ is the length,

in bits, of the description of $Y$ when encoded with $M_{Y|X}$ given the data of $X$ as auxiliary information. Instantiating Equation 2 with MDL, we get the relative amount of directed information from $X$ to $Y$ using MDL, denoted $\hat{\Delta}_{X\to Y}$, as

$$\hat{\Delta}_{X\to Y} = \frac{L(X, M_X) + L(Y, M_{Y|X})}{L(X, M_X) + L(Y, M_Y)} \quad , \tag{3}$$

defining $\hat{\Delta}_{Y\to X}$ analogously. As with Equation 2, $\hat{\Delta}$ takes a value of 1 when $X$ has no information about $Y$. The smaller the value ($< 1$), the more the information $X$ has about $Y$. We infer $X \to Y$, if $\hat{\Delta}_{X\to Y} < \hat{\Delta}_{Y\to X}$. Alternatively, when $\hat{\Delta}_{Y\to X} < \hat{\Delta}_{X\to Y}$, we infer $Y \to X$. When $\hat{\Delta}_{X\to Y} = \hat{\Delta}_{Y\to X}$, we conclude that $X$ and $Y$ are correlated, but do not have a causal relation.

## IV. Causal Inference by Packing Data

In this section we instantiate the above framework for binary data. Henceforth, $X$ represents a binary dataset, and so does $Y$. To infer the causal direction using the MDL-based formulation, we need a causal model class suitable for causal inference on binary dataset. That is, we have to define our model class that allows to causally explain $Y$ given $X$ and vice versa.

There are several ways to capture the causal dependencies. A natural way is to express dependencies in a Directed Acyclic Graph (DAG) where nodes represent the variables and directed edges represent the causal dependencies. This gives us a global model. Alternatively, we can use the decision trees – a form of graphical model – over the variables. With decision trees, not only do we get the global overview, but also we can capture the local dependencies identifying part of the data that may causally depend on the other parts of the data.

We define a *set of decision trees* as our model class. 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 [10]. In particular, we instantiate the MDL-based causal score based on Greedy Pack. Next, we briefly explain how Pack works.

### A. Packing Data

Pack is an MDL based algorithm for discover interesting itemsets from binary data [10]. 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 [10]. While we do not care about these itemsets, it is the decision tree model Pack infers that is of interest to us.

As an example, suppose that we have a binary data with three attributes $a_1$, $a_2$, and $a_3$. Pack aims at discovering the set of trees such that we can encode the whole data in as few as possible bits. In Figure 1(a) – 1(c) we give an example of the trees Pack would find on some hypothetical toy data. As the figure shows, $a_1$ depends on $a_2$, and $a_3$ depends on both $a_1$ and $a_2$. These trees identify both local causal dependencies, as well as the global causal DAG shown in Figure 1(d).

For self containment, let us repeat the main aspects of Pack. Let $n$ be the number of rows in binary data $D$, and $m$, the number of attributes. Suppose we have a model $M$ that consists of a set of decision trees, $M = \{T_1, T_2, \ldots T_m\}$. An attribute $a_i$ is encoded using its decision tree $T_i$, and hence the number of bits needed by $T_i$ to encode $a_i$ over complete data $D$ using optimal Shannon code [14] is given by

$$L_D(T_i) = \sum_{l\in lvs(T_i)} \sum_{v\in\{0,1\}} -n \Pr(a_i = v \mid l) \log \Pr(a_i = v \mid l) \, ,$$

where $lvs(T_i)$ is the set of all leaves of $T_i$, and $\Pr(a_i = v \mid l)$ is the empirical probability of $a_i = v$ given that leaf $l$ is chosen [10]. The outer summation represents the leaves that are selected by testing the values of the other attributes within the same row, whereas the inner summation represents the optimal Shannon code length for encoding two possible values of $a_i$.

The complexity of a leaf $l$ of a tree $T$ is encoded using *refined* MDL [9] as

$$L(l) = \log \sum_{i=0}^{n'} \binom{n'}{i} \left(\frac{i}{n'}\right)^i \left(\frac{n'-k}{n'}\right)^{n'-k} \quad ,$$

where $n'$ is the number of rows for which $l$ is used [10].

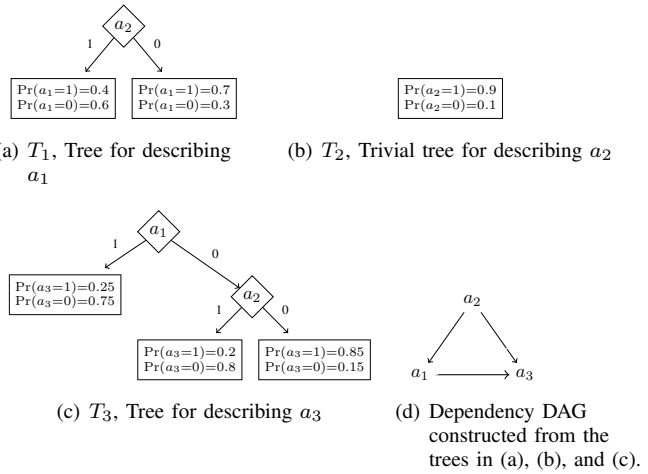
To describe a tree $T_i \in M$, the number of nodes in $T_i$ is encoded first. In doing so, one bit is used to indicate whether the node is a leaf or an intermediate node. For an intermediate node, an extra $\log m$ bits is used to identify the attribute representing the node [10]. Let $intr(T_i)$ be the set of all intermediate nodes of a tree $T_i$. Therefore, the number of bits needed to describe the tree $T_i$ and data $D$ using $T_i$, denoted $L(T_i)$, is given by

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

Putting it together, the total number of bits needed to describe the trees, one for each attribute, and the complete data $D$, denoted $L(D, M)$, is given by

$$L(D, M) = \sum_{T_i\in M} L(T_i) \, . \tag{4}$$

To discover good models directly from data, Tatti & Vreeken propose the Greedy Pack algorithm [10]. Greedy Pack greedily optimises the MDL objective function given in Equation 4. It starts with a tree model consisting only of trivial trees – simplest tree without any other attribute as shown in Figure 1(b). For each attribute, it discovers that split on another attribute that maximises the compression without creating cycles in the model. It greedily accepts the overall best split, and iterates until no further split can be found that saves any bits. We refer the interested reader to the original paper [10] for more details on Pack.




(a) $T_1$, Tree for describing $a_1$

(b) $T_2$, Trivial tree for describing $a_2$

(c) $T_3$, Tree for describing $a_3$

(d) Dependency DAG constructed from the trees in (a), (b), and (c).

Figure 1. In (a), (b), and (c), we give the toy decision trees generated by PACK for a hypothetical binary dataset containing three attributes, namely $a_1$, $a_2$, and $a_3$. In (d), we show the DAG identified by those trees.

### B. Instantiating the MDL-based Score

Next, we discuss how to use PACK to compute the MDL-based causal score. To compute $L(X, M_X)$, we can simply run PACK on $X$. Now, all that remains is to compute $L(Y, M_{Y|X} \mid X)$, i.e. compress $Y$ given $X$. However, PACK does not support conditional compression off-the-shelf. The naive workaround is to compress $X$ followed by $Y$ together to get $L(X, M_X) + L(Y, M_{Y|X} \mid X)$ in one go. However, when doing so, we clearly do not want the attributes in $X$ to depend on the attributes in $Y$. Therefore we modify PACK so that an attribute $x \in X$ considers only the other attributes in $X$ as candidate parents for its decision tree. However, for an attribute $y \in Y$, we consider all the other attributes in $X$ and $Y$ as candidate parents for its decision tree.

To make it fair, we also compute the denominator of Equation 3, i.e. $L(X, M_X)+L(Y, M_Y)$ in one go. For that we modify PACK so that it only considers the other attributes in $X$ when constructing a decision tree for an attribute in $X$, and it only considers the other attributes in $Y$ when constructing a decision tree for an attribute in $Y$. From here onwards, we refer to the PACK-based instantiation of Equation 3 as ORIGO, which means *origin* in latin. Further, we refer to the PACK-based instantiation of Equation 1 as ORIGI (nothing special about the name here).

Although our focus is primarily on binary data, we can infer causal direction from categorical data as well. To this end, we can binarise 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 discretisation, we can infer reliable causal directions from continuous real-valued data as well.

### C. Computational Complexity

Next, we analyse the computational complexity of ORIGO. To compute $\hat{\Delta}_{X \to Y}$, we have to run PACK on the full dataset twice – once for denominator, and once for numerator in Equation 3. 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 $Y$. To infer the causal direction, we have to compute both $\hat{\Delta}_{X \to Y}$, and $\hat{\Delta}_{Y \to X}$. Therefore, in the worst case, the computational complexity of ORIGO is $\mathcal{O}(2^m n)$. Although this looks horrible, in practice, ORIGO is fast, and completes within seconds.

## V. RELATED WORK

Inferring causal direction from observational data is a challenging task because of the lack of controlled randomised experiments. However, it has also attracted quite a lot of attention over the years [1], [2], [15], [16]. Yet, most of the causal inference frameworks are built for either continuous real-valued, or discrete numeric data.

Constraint-based approaches like conditional independence test [1], [15] require at least three observed random variables. Moreover, these constraint-based approaches cannot distinguish Markov equivalent causal DAGs [17] and therefore 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 properties of the joint distribution. In particular, the Additive Noise Models (ANMs) [4], [16], [18], [19] 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. [20] proposes an ANM for discrete numeric data. These methods, however, assume the joint probability distribution, the class of causal dependencies, as well as the distribution of the noise. Moreover, as they rely on regression, it is not straightforward to adopt ANMs for modeling ordinal or nominal variables.

Further, there are methods that rely on the asymmetry of the joint distribution to distinguish the cause from the effect. The linear trace method [21], [22] 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 $X$, $\Sigma_X$. The kernelized trace method [23] can infer non-linear causal relations, but requires the causal relation to be deterministic, functional, and invertible. In contrast, we do not make any assumptions on the causal relation between the variables.

The algorithmic information-theoretic approach views causality in terms of the algorithmic similarity between objects. The key idea is that if $X$ causes $Y$, the shortest description of the joint distribution $P(X, Y)$ is given by the separate descriptions of the distributions $P(X)$ and $P(Y \mid X)$ [2]. It has also been used in justifying the additive noise model based causal discovery [24].

However, since the Kolmogorov complexity is not computable, causal inference using algorithmic information theoretic frameworks require practical implementations, or notions

of independence. For instance, the information-geometric approach [3] defines independence via orthogonality in information space. Janzing & Schölkopf [2] sketch how comparing marginal distributions, and resource bounded computation could be used to infer causation, but do not give practical instantiations. Vreeken [5] propose a causal framework and instantiates it with the cumulative entropy to infer the causal direction in continuous real-valued data.

All above methods consider numeric data only. Causal inference on observational binary or nominal data has seen much less attention. The classic proposal by Silverstein et al. [25] relies on a conditional independence test, and hence require an independent variable $Z$ to be able to determine whether $X$ and $Y$ are causally related. Our closest competitor is the very recent proposal by Liu et al. [26]. DC uses distance correlation between empirical distributions $P(X)$ and $P(Y \mid X)$ to infer the causal direction from multivariate categorical data. In contrast, our method does not only provide the strength of causal direction, but as the experiments show its decision tree based model allows it to recognize more subtle dependencies.

## VI. EXPERIMENTS

We implemented ORIGO in Python and provide the source code for research purposes, along with the used datasets, and synthetic dataset generator.[2] 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. In particular, we note that both ORIGO, and ORIGI are parameter-free. We compare ORIGO against ORIGI, and DC [26].

### A. 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 $Y$ depend on the attributes in $X$ with certain probability, termed here onwards as *dependency*. Throughout the experiments on synthetic data, we generate $X$ of size $n$-by-$k$, and $Y$ of size $n$-by-$l$. In particular, we generate both $X$, and $Y$ with $n = 5000$ rows.

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.

This way, we are certain that there is a dependency in one direction. In general, we expect this direction to be the true causal direction, i.e. $X \rightarrow Y$. Note that we cannot be absolutely sure that the model in the reverse direction, from $Y$ to $X$, would be inferior to the one that we plant all the time. All the reported values are averaged over 200 samples unless stated otherwise.

*1) 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 the trees with the maximum possible height, i.e. $k + l - 1 = 5$. In Figure 2(a), 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 2(b), 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 can already infer the correct causal direction when there are only few causal edges in the DAG.

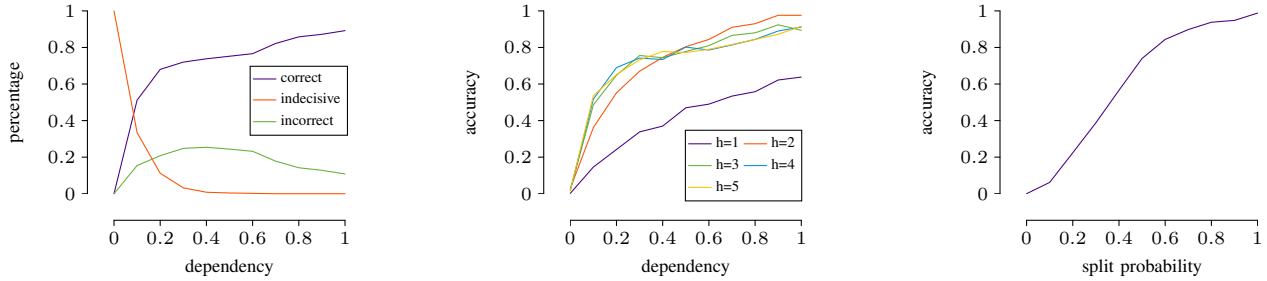
Next, we analyse the effect of the split probability on the accuracy of ORIGO. For that, we set $k = l = 3$, the dependency to 1.0, and generate trees with the maximum possible height. In Figure 2(c), we observe that the accuracy of ORIGO increases with the increase in the split probability. This is 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 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 200 cause-effect pairs where 100 of them have $k = 1, l = 3$ and remaining 100 have $k = 3, l = 1$. In Figure 4(a), we give the plot comparing the accuracy of ORIGO against ORIGI and DC. We see that ORIGO outperforms both competitors by a fair margin at every dependency. The difference in accuracy gets larger as the dependency increases. Notably, this observation also empirically bolsters our argument on the unbiased nature of the relative joint complexity in Section III.

Next, we consider the symmetric case where $k = l = 3$. For the experiment we use set the split probability to 1.0, and generate trees with the maximum possible height. In Figure 4(b), we show the plot comparing the accuracy of ORIGO against ORIGI, and DC. We see that both ORIGO and ORIGI outperform DC at almost every dependency. We note that for the pairs without dependency, DC infers a causal relationship in over 45% of the cases.

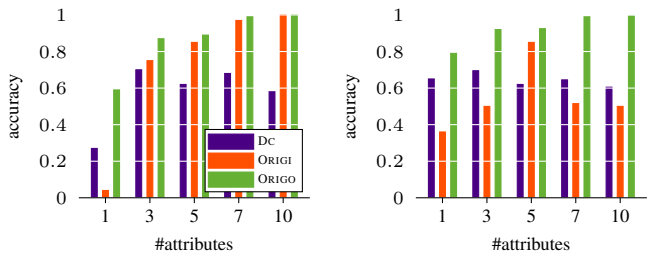
*2) Dimensionality:* Next, we study the robustness against data dimensionality. At first, we consider cause-effect pairs

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




(a) depdency vs various metrics (b) dependency vs accuracy (c) split probability vs accuracy

Figure 2. For synthetic datasets with $k = l = 3$, we report (a) various metrics at various dependencies (b) the accuracy at various dependencies for trees with various maximum heights, and (c) the accuracy at various split probabilities for ORIGO.




(a) symmetric case, $k = l$ (b) asymmetric case

Figure 3. For synthetic datasets, we report the accuracy (a) in symmetric case with $k = l$, and (b) in asymmetric case (5 vs. varying cardinalities).

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 ORIGI and DC. In Figure 3(a), we see that ORIGO obtains high accuracy in every setting. With the exception of the univariate case, ORIGI 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 ORIGI by a huge margin the stronger the unbalance between the cardinalities of $X$ and $Y$. This is explained by the inherent bias of ORIGI as discussed in Section III. In addition, we see that ORIGO outperforms DC in every setting.

*3) Hypothesis Testing:* To evaluate whether ORIGO infers relevant causal relationship, we employ swap randomisation [27]. Let $\epsilon$ denote the absolute difference between the directed amount of information from $X$ to $Y$ and vice versa. That is, $\epsilon = |\hat{\Delta}_{X \to Y} - \hat{\Delta}_{Y \to X}|$. We compare the $\epsilon$ value of the actual cause-effect pair to those of 100 swap randomised versions of the pair. We set $k = l = 3$, the dependency to 1.0, the probability of split to 1.0, and generate trees with maximum possible height. The null hypothesis is that the $\epsilon$ value of the actual data is likely to occur in random data. However, we observe that the probability of getting a $\epsilon$ 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.

To assess whether ORIGO infers causal relationship when the 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 100 cause-effect pairs with no causal relationship. Then we compute their $\epsilon$ values and set the cut-off $\epsilon$ value at a significance level of 0.05. Next, we generate new 100 cause-effect pairs with causal relationship. The statistical power is the proportion of the 100 new cause-effect pairs whose $\epsilon$ 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 4(c). The lines corresponding to ORIGO and ORIGI overlap as both have the same high statistical power, outperforming DC in every setting.

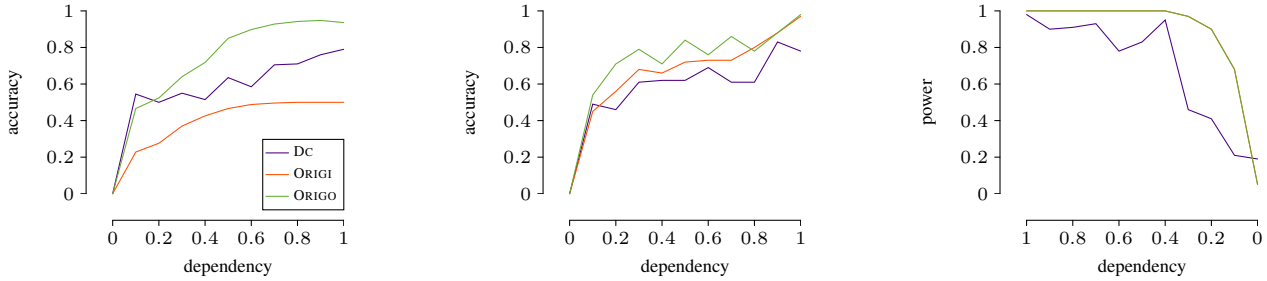
Last, but not least, we observe that for all the above experiments inferring the causal direction for one pair typically takes only up to a few seconds.

Next, evaluate ORIGO on real-world data.

### B. Real-world Data

*1) Univariate Pairs:* First, we evaluate ORIGO on benchmark cause-effect pairs with known ground truth [28]. In particular, we take 95 univariate pairs. We considered various discretisation strategies – including Equi-Frequency, and Equi-Width binning, MDL-based histogram density estimation [29], and parameter-free unsupervised interaction preserving discretisation (IPD) [30]. We obtained the best results using IPD, and will report these below.

In general, we can trade-off the percentage of correct decisions versus the percentage of cases in which a decision is taken (*decision rate*) by taking decisions only for $\epsilon \geq \epsilon_t$ for some threshold $\epsilon_t$. Following [3], we show the percentage of correct decisions versus the decision rate in Figure 5. If we look over all the pairs, we find that ORIGO infers correct direction in roughly 58% of all decisive pairs. When we consider only those pairs where $\epsilon$ is relatively high, i.e. those




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

Figure 4. For synthetic datasets, we compare (a) the accuracy in asymmetric case (1 vs. 3), (b) the accuracy at various dependencies in symmetric case ($k = l = 3$), (c) the statistical power at various dependencies.

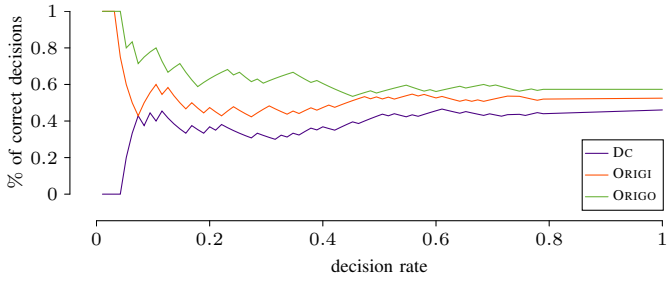



Figure 5. Percentage of correct decisions vs decision rate for univariate Tübingen cause-effect pairs discretised using IPD.

Table I
RESULTS ON MULTIVARIATE CAUSE-EFFECT PAIRS WITH KNOWN GROUND TRUTH. "✓" MEANS THE CORRECT CAUSAL DIRECTION IS INFERRED, "×" MEANS THE WRONG DIRECTION, AND "−" MEANS INDECISION.

| Dataset | #rows | $\|X\|$ | $\|Y\|$ | Truth | ORIGO | ORIGI | 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 | $Y \to X$ | ✓ | ✓ | − |

pairs where ORIGO is most decisive we see that over the top 10% most decisive pairs it is 80% accurate, yet still 70 % accurate for the top 30 % pairs, which is on-par with the top-performing causal inference frameworks for continuous real-valued data [3], [4].

*2) 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 [28]. We use IPD to discretise the data. We give the base statistics in Table I. 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, ORIGI, and DC.

We find that ORIGO infers correct direction in 3 pairs and is indecisive in *Weather forecast* dataset. On the other hand, ORIGI infers correct direction in all datasets. DC, however, is either indecisive or incorrect.

### C. Qualitative Results

*1) Acute inflammation dataset:* The *Acute inflammations* dataset is taken from the UCI repository.[3] 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 ($y_2$).

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 $X, Y$ where $X = \{x_{11}, x_{12}, x_3, x_4, x_5, x_6\}$ and $Y = \{y_1, y_2\}$. We find that $Y \to X$. That is, ORIGO infers that the diseases cause the symptoms, which is in agreement with intuition.

*2) ICDM abstracts dataset:* Next, we consider the *ICDM abstracts* dataset, which is available from the authors of [31]. 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 [32]. Then, we apply ORIGO on those 100 self-sufficient itemsets. We sort the discovered causal directions by their $\epsilon$ value in decreasing order. In Table II, we give 14 highly characteristic and non-redundant results along with their $\epsilon$ values taken from top 24 causal directions. We expect the causal directions having higher $\epsilon$ 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 areas of data mining and studied by many. Clearly, when *frequent itemset* appears in a text, it is likely to cause *mining* to appear in the text than vice versa. Likewise, *neural* is likely to cause *network* to appear in the text and not the other way around.

[3]http://archive.ics.uci.edu/ml/

Table II
RESULTS OF ORIGO ON *ICDM*. WE SORT THE DISCOVERED CAUSAL DIRECTIONS USING THEIR $\epsilon$ VALUES IN DESCENDING ORDER AND GIVE 14 CHARACTERISTIC AND NON-REDUNDANT EXEMPLARS DRAWN FROM TOP 24 CAUSAL DIRECTIONS.

| discovered causal direction | $\epsilon$ | discovered causal direction | $\epsilon$ |
|---|---|---|---|
| *frequent itemset* → *mining* | 0.002966 | *lda* → *discriminant* | 0.001553 |
| *drift* → *concept* | 0.002048 | *neural* → *network* | 0.001444 |
| *walk* → *random* | 0.001912 | *edge* → *graph* | 0.001426 |
| *lda* → *linear* | 0.001797 | *social* → *network* | 0.001312 |
| *upper* → *bound* | 0.001730 | *subgraph* → *graph* | 0.001281 |
| *fp* → *tree* | 0.001677 | *collaborative* → *filtering* | 0.001029 |
| *anomaly* → *detection* | 0.001599 | *outlier* → *detection* | 0.001001 |

Overall, we see that the causal directions discovered by ORIGO in the ICDM dataset are sensible.

*3) Adult dataset:* 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 binary attributes as "private", "self-employed", "public-servant", and "unemployed". We binarise *education* attribute into seven binary attributes as "dropout", "associates", "bachelors", "doctorate", "hs-graduate", "masters", and "prof-school". Further, we binarise *occupation* attribute into eight binary attributes as "admin", "armed-force", "blue-collar", "white-collar", "service", "sales", "professional", and "other-occupation". Lastly, we binarise *income* attribute into two binary attributes as "*>50K*" and "*<=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 III, we report 7 interesting and non-redundant causal directions identified by ORIGO drawn from the top 19 strongest causal directions. Inspecting the results, we see that ORIGO infers sensible causal directions from the *adult* dataset. For instance, take these two causal directions *public-servant professional masters* → *>50K* and *public-servant admin hs-graduate* → *<=50K*. Whereas a professional with a master's degree working in a public office is likely to earn *>50K*, a high school graduate working in a public office in an administrative position is likely to earn *<=50K*.

Overall, these results show that ORIGO finds sensible causal directions from real-world data.

Table III
RESULTS OF ORIGO ON *Adult*. WE GIVE 7 CHARACTERISTIC EXEMPLARS DRAWN FROM THE TOP RANKED CAUSAL DIRECTIONS.

| discovered causal direction | $\epsilon$ |
|---|---|
| *public-servant professional doctorate* → *>50K* | 0.000126 |
| *self-employed white-collar* → *>50K* | 0.000108 |
| *public-servant professional masters* → *>50K* | 0.000108 |
| *public-servant admin hs-graduate* → *<=50K* | 0.000107 |
| *blue-collar dropout* → *<=50K private* | 0.000103 |
| *bachelors self-employed white-collar* → *>50K* | 0.000096 |
| *admin hs-graduate* → *<=50K* | 0.000092 |

## VII. 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 sub-optimal discretization in the pre-processing. 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 binarising 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 discretisation strategies to discretise continuous real-valued data. Different discretisation strategies yield different outcomes. Consequently, we observe different results on continuous real-valued data depending on the discretisation strategy we pick. It would make an engaging future work to devise a discretisation 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.

Note that 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.

## VIII. 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 state-of-the-art methods by a wide margin.

## ACKNOWLEDGEMENTS

Kailash Budhathoki is supported by the International Max Planck Research School for Computer Science (IMPRS-CS). Both authors are supported by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

## REFERENCES

[1] J. Pearl, *Causality: Models, Reasoning, and Inference*. New York, NY, USA: Cambridge University Press, 2000.

[2] D. Janzing and B. Schölkopf, "Causal inference using the algorithmic markov condition," *IEEE Transactions on Information Technology*, vol. 56, no. 10, pp. 5168–5194, 2010.

[3] D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf, "Information-geometric approach to inferring causal directions," *Artificial Intelligence*, vol. 182-183, pp. 1–31, 2012.

[4] J. Peters, J. Mooij, D. Janzing, and B. Schölkopf, "Causal discovery with continuous additive noise models," *Journal of Machine Learning Research*, vol. 15, pp. 2009–2053, 2014.

[5] J. Vreeken, "Causal inference by direction of information," in *Proceedings of the 15th SIAM International Conference on Data Mining (SDM), Vancouver, Canada*, 2015, pp. 909–917.

[6] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems of Information Transmission*, vol. 1, pp. 1–7, 1965.

[7] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*. Springer, 1993.

[8] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 1, pp. 465–471, 1978.

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

[10] N. Tatti and J. Vreeken, "Finding good itemsets by packing data," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy*, 2008, pp. 588–597.

[11] R. J. Solomonoff, "A formal theory of inductive inference. part I, II," *Information and Control*, vol. 7, pp. 1–22224–254, 1964.

[12] G. J. Chaitin, "On the simplicity and speed of programs for computing infinite sets of natural numbers," *Journal of the ACM*, vol. 16, no. 3, pp. 407–422, 1969.

[13] P. Grünwald and P. M. B. Vitányi, "Shannon information and Kolmogorov complexity," *CoRR*, vol. cs.IT/0410002, 2004.

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

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

[16] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, pp. 2003–2030, 2006.

[17] T. Verma and J. Pearl, "Equivalence and synthesis of causal models," in *Proceedings of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI)*, 1991, pp. 255–270.

[18] P. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Schölkopf, "Nonlinear causal discovery with additive noise models," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2009, pp. 689–696.

[19] K. Zhang and A. Hyvärinen, "On the identifiability of the post-nonlinear causal model," in *Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009, pp. 647–655.

[20] J. Peters, D. Janzing, and B. Schölkopf, "Identifying cause and effect on discrete data using additive noise models," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 597–604.

[21] D. Janzing, P. Hoyer, and B. Schölkopf, "Telling cause from effect based on high-dimensional observations," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2010, pp. 479–486.

[22] J. Zscheischler, D. Janzing, and K. Zhang, "Testing whether linear equations are causal: A free probability theory approach," *CoRR*, vol. abs/1202.3779, 2012.

[23] Z. Chen, K. Zhang, and L. Chan, "Nonlinear causal discovery for high dimensional data: A kernelized trace method," in *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM)*, 2013, pp. 1003–1008.

[24] D. Janzing and B. Steudel, "Justifying additive noise model-based causal discovery via algorithmic information theory," *Open Systems and Information Dynamics*, vol. 17, no. 2, pp. 189–212, 2010.

[25] C. Silverstein, S. Brin, R. Motwani, and J. Ullman, "Scalable techniques for mining causal structures," *Data Mining and Knowledge Discovery*, vol. 4, no. 2, pp. 163–192, 2000.

[26] F. Liu and L. Chan, "Causal inference on discrete data via estimating distance correlations," *Neural Computation*, vol. 28, no. 5, pp. 801–814, 2016.

[27] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas, "Assessing data mining results via swap randomization," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 3, pp. 167–176, 2007.

[28] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf, "Distinguishing cause from effect using observational data: Methods and benchmarks," *Journal of Machine Learning Research*, vol. 17, no. 32, pp. 1–102, 2016.

[29] P. Kontkanen and P. Myllymäki, "MDL histogram density estimation," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS), San Juan, Puerto Rico*, 2007.

[30] H. V. Nguyen, E. Müller, J. Vreeken, and K. Böhm, "Unsupervised interaction-preserving discretization of multivariate data," *Data Mining and Knowledge Discovery*, vol. 28, no. 5-6, pp. 1366–1397, 2014.

[31] T. De Bie, "Maximum entropy models and subjective interestingness: an application to tiles in binary databases," *Data Mining and Knowledge Discovery*, vol. 23, no. 3, pp. 407–446, 2011.

[32] G. Webb, "Filtered-top-k association discovery," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 183–192, 2011.