Knowledge and Information Systems
https://doi.org/10.1007/s10115-018-1286-7

REGULAR PAPER

Telling cause from effect by local and global regression

Alexander Marx[1] · **Jilles Vreeken**[1]

Received: 1 February 2018 / Revised: 5 September 2018 / Accepted: 24 November 2018

Abstract
We consider the problem of inferring the causal direction between two univariate numeric random variables X and Y from observational data. This case is especially challenging as the graph X causes Y is Markov equivalent to the graph Y causes X, and hence it is impossible to determine the correct direction using conditional independence tests. To tackle this problem, we follow an information theoretic approach based on the algorithmic Markov condition. This postulate states that in terms of Kolmogorov complexity the factorization given by the true causal model is the most succinct description of the joint distribution. This means that we can infer that X is a likely cause of Y when we need fewer bits to first transmit the data over X, and then the data of Y as a function of X, than for the inverse direction. That is, in this paper we perform causal inference by compression. To put this notion to practice, we employ the Minimum Description Length principle, and propose a score to determine how many bits we need to transmit the data using a class of regression functions that can model both local and global functional relations. To determine whether an inference, i.e. the difference in compressed sizes, is significant, we propose two analytical significance tests based on the no-hypercompression inequality. Last, but not least, we introduce the linear-time SLOPE and SLOPER algorithms that through thorough empirical evaluation we show outperform the state of the art by a wide margin.

Keywords Causal inference · Regression · MDL · Kolmogorov complexity

1 Introduction

Telling apart cause and effect given only observational data is one of the fundamental problems in science [22,31]. We consider the problem of inferring the most likely causal direction between two statistically dependent univariate numeric random variables X and Y, given only a sample from their joint distribution, and assuming no hidden confounder Z causing both

✉ Alexander Marx
amarx@mpi-inf.mpg.de

Jilles Vreeken
jilles@mpi-inf.mpg.de

[1] Max Planck Institute for Informatics and Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

Published online: 07 December 2018

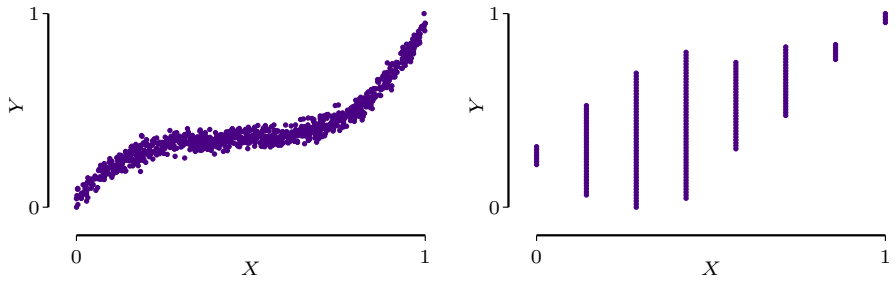
X and Y. That is, we are interested in identifying whether X causes Y, whether Y causes X, or whether they are merely correlated.

Traditional methods, that rely on conditional independence tests, cannot decide between the Markov equivalent classes of $X \to Y$ and $Y \to X$ [22], as these result in the same joint distribution. Recently, it has been postulated that if $X \to Y$, there exists an independence between the marginal distribution of the cause, $P(X)$, and the conditional distribution of the effect given the cause, $P(Y \mid X)$ [10,30]. Intuitively, we assume X to be generated 'first' and Y to be generated by some process that takes both X and noise independent of X as inputs. This means that in the true causal direction the distribution of Y given X will just be the distribution of the noise, which is independent of X. In the anti-causal direction, however, such an independence does not hold; to reconstruct X from Y, we needed to remove the noise, and hence $P(Y)$ and $P(X \mid Y)$ remain dependent. The state of the art exploits this asymmetry in various ways and overall obtain up to 70% accuracy on a well-known benchmark of cause–effect pairs [9,12,21,25,29]. In this paper, we break this barrier and give an elegant score that is computable in linear time and obtains over 82% accuracy on the same benchmark.

We base our method on the algorithmic Markov condition, a recent postulate by Janzing and Schölkopf [10], which states that if X causes Y, the factorization of the joint distribution $P(X, Y)$ in the causal direction has a simpler description—in terms of Kolmogorov complexity—than that in the anti-causal direction. That is, if $X \to Y$, $K(P(X)) + K(P(Y \mid X)) \leq K(P(Y)) + K(P(X \mid Y))$. The key idea is strongly related to that above. Namely, because the distribution of the cause and the distribution of the effect conditioned on the cause are independent, we do not lose any bits compared to the optimal compression if we describe these two terms separately. In the anti-causal direction, however, because $P(X \mid Y)$ is dependent on $P(Y)$, we have to 'tune' the noise and hence have to spend additional bits that are not needed in the causal direction. As any physical process can be modelled by a Turing machine, this ideal score can detect any causal dependence that can be explained by a physical process. However, Kolmogorov complexity is not computable, so we need a practical instantiation of this ideal. In this paper, we do so using the Minimum Description Length (MDL) principle, which provides a statistically well-founded approximation of Kolmogorov complexity.

Simply put, we propose to fit a regression model from X to Y, and vice versa, measuring both the complexity of the function, as well as the error it makes in bits, and infer that causal direction by which we can describe the data most succinctly. We carefully construct an MDL score such that we can meaningfully compare between different types of functional dependencies, including linear, quadratic, cubic, reciprocal and exponential functions, and the error that they make. This way, for example, we will find that we can more succinctly describe the data in Fig. 1a by a cubic function than with a linear function, as while it takes fewer bits to describe the latter function, it will take many more bits to describe the large error it makes.

We do not only consider models that try to explain all the data with a single, global, deterministic regression function, but also allow for non-deterministic models. That is, we consider compound regression functions that extend the global deterministic function by also including regression functions for local parts of the data corresponding to specific, duplicated X values. For example, consider the data in Fig. 1b, where the Y values belonging to a single X value clearly show more structure than the general linear trend. In contrast, if we rotate the plot by 90°, we do not observe the same regularities for the X values mapped to a single Y value. In many cases, e.g. $Y = 1$ there is only one mapping X value. We can exploit this asymmetry by considering local regression functions per value of X, each individually fitted but as we assume all non-deterministic functions to be generated by the same process, all



(a) deterministic **(b)** non-deterministic

Fig. 1 Example deterministic and non-deterministic data. In both cases, the ground truth is X causes Y. The left-hand data are generated using a cubic function with Gaussian noise, whereas the right-hand data are generated using a non-deterministic function

should be of the same function class. In this particular example, we therewith correctly infer that X causes Y. The MDL principle prevents us from overfitting, as such local functions are only included if they aid global compression. Last, but not least, we give a linear-time algorithm, SLOPE, to compute this score.

As we model Y as a function of X and noise, our approach is somewhat reminiscent to causal inference based on Additive Noise Models (ANMs) [30], where one assumes that Y is generated as a function of X plus additive noise, $Y = f(X) + N$ with $X \perp\!\!\!\perp N$. In the ANM approach, we infer $X \to Y$ if we can find a function from X to Y that admits an ANM, but cannot do so in the opposite direction. In practice, ANM methods often measure the independence between the presumed cause and the noise in terms of p values, and infer the direction of the lowest p value. As we will see, this leads to unreliable confidence scores—not the least because p values are often strongly influenced by sample size [1], but also as that a lower p value does not necessarily mean that H_1 is more true, just that H_0 is very probably not true [1]. We will show that our score, on the other hand, is robust against sample size, and correlates strongly with accuracy. Moreover, it admits an elegant and effective analytical statistical test on the *difference* in score between the two causal directions based on the no-hypercompression inequality [4,8].

Our key contributions can be summarized as follows, we

(a) show how to model unobserved mechanisms via compound deterministic and non-deterministic functions,
(b) propose an MDL score for causal inference on pairs of univariate numeric random variables,
(c) formulate two analytic significance tests based on compression,
(d) introduce the linear-time algorithms SLOPE and SLOPER,
(e) give extensive empirical evaluation, including a case study
(f) and make all code, data generators and data available.

This paper builds upon and extends the work appearing in ICDM'17 [18]. Notably, we provide a link between the confidence and significance score of our method. In addition, we derive a second, p value test relative to the sample size. This new test allows us to set a threshold directly for the confidence value. To improve the generality of our inference algorithm, we include more basis functions and allow combinations of them. As a result, we propose SLOPER, which can fit more complex functions, if necessary. Further, we provide theory to link the identifiability of our approach to ANMs and discuss to which extend this

holds. Last, we give a more thorough evaluation of SLOPE and SLOPER on synthetic and real data and include results with respect to identifiability of ANMs on synthetic data.

The remainder of this paper is organized as usual. We first give a brief primer to Kolmogorov complexity and the Minimum Description Length principle in Sect. 2. In Sect. 3, we introduce our score based on the algorithmic independence of conditional, as well as a practical instantiation based on the MDL principle. Section 4 rounds up the theory by discussing identifiability and significance tests. To efficiently compute this score, we introduce the linear-time SLOPE and SLOPER algorithms in Sect. 5. Section 6 discusses related work. We empirically evaluate our algorithms in Sect. 7 and discuss the results in Sect. 8. We round up with conclusions in Sect. 9.

2 Preliminaries

In causal inference, the goal is to determine for two random variables X and Y that are statistically dependent whether it is more likely that X *causes* Y, denoted by $X \to Y$, or whether it is more likely that Y causes X, $Y \to X$. In this paper, we consider the case where X and Y are univariate and numeric. We work under the common assumption of causal sufficiency [5,21,25,33]. That is, we assume there is no hidden confounder variable Z that causes both X and Y.

We base our causal inference score on the notion of Kolmogorov complexity, which we will approximate via the Minimum Description Length (MDL) principle. Below we give brief primers to these two main concepts.

2.1 Kolmogorov complexity

The Kolmogorov complexity of a finite binary string x is the length of the shortest binary program p^* for a universal Turing machine $\mathcal{U}$ that outputs x and then halts [13,15]. Formally,

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

Simply put, p^* is the most succinct *algorithmic* description of x, and therewith Kolmogorov complexity of x is the length of its ultimate lossless compression. Conditional Kolmogorov complexity, $K(x \mid y) \leq K(x)$, is then the length of the shortest binary program p^* that generates x, and halts, given y as input.

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 $\langle x, q\rangle$ [15]. More formally, we have

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

The conditional, $K(P \mid Q)$, is defined similarly except that the universal Turing machine $\mathcal{U}$ now gets the additional information Q. The algorithmic mutual information between two distributions P and Q is $I(P : Q) = K(P) - K(P \mid Q^*)$, where Q^* is the shortest binary program for Q. For more details on Kolmogorov complexity, see [15].

2.2 Minimum Description Length principle

Kolmogorov complexity is not computable [15]. We can, however, approximate it from above through lossless compression [15]. The Minimum Description Length (MDL) principle [8,27]

provides a statistically well-founded and computable framework to do so. Conceptually, instead of all programs, *Ideal MDL* considers only those for which we know that they output x and halt, i.e. lossless compressors. Formally, given a model class $\mathcal{M}$, MDL identifies the best model $M \in \mathcal{M}$ for data D as the one minimizing

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

where $L(M)$ is the length in bits of the description of M, and $L(D \mid M)$ is the length in bits of the description of data D given M. This is known as a two-part or *crude* MDL. There also exists one-part, or *refined* MDL. Although refined MDL has theoretically appealing properties, it is only efficiently computable for a small number of model classes.

To use MDL in practice we both need to define a model class and how to encode a model and the data given a model in bits. It is important to note that in both Kolmogorov complexity and MDL we are only concerned with optimal code *lengths*, not actual codes—our goal is to measure the *complexity* of a data set under a model class, after all [8]. As is usual in MDL, all logarithms are to base 2, and we use the common convention that $0 \log 0 = 0$.

3 Information theoretic causal inference

In this section, we first introduce how to infer causal directions using Kolmogorov complexity. Thereupon, we show how to obtain a computable score based on the MDL principle.

3.1 Causal inference by Kolmogorov complexity

A central postulate in causal inference concerns the algorithmic independence of conditionals. For multiple random variables, this postulate is defined as follows [10].

Algorithmic independence of conditionals: *A causal hypothesis is only acceptable if the shortest description of the joint density P is given by the concatenation of the shortest description of the Markov kernels. Formally, we write*

$$K(P(X_1, \ldots, X_n)) \stackrel{+}{=} \sum_j K(P(X_j \mid PA_j)), \tag{1}$$

which holds up to an additive constant independent of the input, and where PA_j corresponds to the parents of X_j in a causal directed acyclic graph (DAG).

As we consider two variables, X and Y, either X is the parent of Y or the other way round. That is, either

$$K(P(X, Y)) \stackrel{+}{=} K(P(X)) + K(P(Y \mid X)) \text{ , or}$$
$$K(P(X, Y)) \stackrel{+}{=} K(P(Y)) + K(P(X \mid Y)).$$

In other words, two valid ways to describe the joint distribution of X and Y include to first describe the marginal distribution $P(X)$ and then the conditional distribution $P(Y \mid X)$, or first to describe $P(Y)$ and then $P(X \mid Y)$.

Thereupon, Janzing and Schölkopf formulated the postulate for algorithmic independence of Markov kernels [10].

Algorithmic independence of Markov kernels: *If $X \to Y$, the marginal distribution of the cause $P(X)$ is algorithmically independent of the conditional distribution of the effect given the cause $P(Y \mid X)$, i.e. the algorithmic mutual information between the two will be zero,*

$$I(P(X) : P(Y \mid X)) \stackrel{+}{=} 0, \tag{2}$$

while this is not the case in the other direction.

Simply put, for the true causal direction, the marginal distribution of the cause is algorithmically independent of the conditional distribution of the effect given the cause. Building upon Eqs. (1) and (2), Mooij et al. [20] derived an inference rule stating that if X causes Y,

$$K(P(X)) + K(P(Y \mid X)) \leq K(P(Y)) + K(P(X \mid Y)) \tag{3}$$

holds up to an additive constant. This means that if $X \to Y$, the description of the joint distribution $K(P(X, Y))$ of first describing the marginal distribution of the cause $K(P(X))$ and then describing the conditional distribution of the effect given the cause $K(P(Y \mid X))$ will be shorter than the other way around.

Although Eq. (3) already allows for inferring the causal direction for a given pair, we obtain a more robust score, allowing for fair comparison of results independent of data sizes, when we normalize the result. In particular, Budhathoki and Vreeken [5] recently proposed to normalize the scores with the sum of the description lengths for the marginal distributions. We therefore define our causal indicator as

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

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

The confidence of our score is $\mathbb{C} = |\Delta_{X \to Y} - \Delta_{Y \to X}|$. The higher, the more certain we are that the inferred causal direction is correct. To avoid confusion, we want to emphasize that $\mathbb{C}$ has nothing to do with a confidence interval, but can be used to rank results of several tests. Below, after introducing our practical score, we will show how we can in addition define two analytical tests to determine whether an inference is statistically significant.

3.2 Causal inference by MDL

As Kolmogorov complexity is not computable, we will instantiate $\Delta_{X \to Y}$ and $\Delta_{Y \to X}$ using the Minimum Description Length principle [8,15]. In practice, this means we will estimate $\Delta_{X \to Y}$ as

$$\hat{\Delta}_{X \to Y} = \frac{L(X) + L(Y \mid X)}{L(X) + L(Y)}$$

where $L(X)$ is the length in bits of the description of the marginal distribution of X, $L(Y)$ that of the marginal distribution of Y, and $L(Y \mid X)$ that of the conditional distribution of Y given X. We define $\hat{\Delta}_{Y \to X}$ analogue to $\hat{\Delta}_{X \to Y}$, and we infer $X \to Y$, if $\hat{\Delta}_{X \to Y} < \hat{\Delta}_{Y \to X}$, $Y \to X$, if $\hat{\Delta}_{X \to Y} > \hat{\Delta}_{Y \to X}$ and do not decide if $\hat{\Delta}_{X \to Y} = \hat{\Delta}_{Y \to X}$ or below a user-defined threshold. Like above, confidence $\mathbb{C}$ is simply the absolute difference between $\hat{\Delta}_{X \to Y}$ and $\hat{\Delta}_{Y \to X}$.

Considering the difference between the encoded lengths is related to, but not the same as considering the ratio of the posteriors; we also include the complexity of the model,

which helps against overfitting. Intuitively, if the functions we find for the two directions both explain the data equally well, we prefer that direction that explains it using the simplest function.

This leaves us to explain how we encode the data, and, most importantly, how we encode $L(Y \mid X)$.

Intuition of the conditional encoding

The general idea is simple: we use regression to model the data of Y given X. That is, we model Y as a function f of X and independent noise N, i.e. $Y = f(X) + N$. We do so by fitting a regression function f over X and treating the error it makes as Gaussian distributed noise. Naturally, the better $f(X)$ fits Y, the fewer bits we will have to spend on encoding errors. The more parameters $f(X)$ have, however, the more bits we will have to spend on encoding these. This way, MDL naturally balances the complexity of the model to that of the data [8]. For example, while a linear function is more simple to describe than a cubic one, the latter will fit the data plotted in Fig. 1a so much better that MDL decides it is the better choice.

A key idea in our approach is to consider not only single global deterministic regression functions f_g, which works well for deterministic data, but to also non-deterministic, or compound functions as models. That is, we consider models that besides the global regression function f_g may additionally consist of *local* regression functions f_l that model Y for those values x of X that non-deterministically map to multiple values of Y. That is, per such value of X, we take the associated values of Y, sort these ascending and uniformly re-distribute them on X over a fixed interval. We now see how well, just for these re-distributed points, we can fit a local regression model f_l. This way, we will for example be able to much more succinctly describe the data in Fig. 1b than with a single global deterministic regression function, as we can now exploit the structure that the values of Y have given a value of X, namely being approximately equally spaced. To avoid overfitting, we use MDL, and only allow a local function for a value of X into our model if it provides a gain in overall compression. Since we assume that for the true causal model the data in the local components follow the same pattern, we only allow models in which all local functions are of the same type, e.g. all are linear, all are quadratic.

In the following paragraphs, we formalize these ideas and define our cost functions.

Complexity of the marginals

We start by defining the cost for the marginal distributions, $L(X)$ and $L(Y)$, which mostly serve to normalize our causal indicators $\hat{\Delta}_{X \to Y}$ and $\hat{\Delta}_{Y \to X}$. As we beforehand do not know how X or Y are distributed, and do not want to incur any undue bias, we encode both using a uniform prior with regard to the data resolution τ of X and Y. That is, we have $L(X) = -n \log \tau_X$, where τ is the resolution of the data of X. Note that resolution τ can be different between X and Y—we specify how we choose τ in the next section. We define $L(Y)$ analogue.

Complexity of the conditional model

Formally, we write F for the set of regression functions, or model, we use to encode the data of Y given X. A model F consists of at least one global regression function $f_g \in \mathcal{F}$, and up

to the size of the domain of X local regression functions $f_l \in \mathcal{F}$, associated with individual values of X. We write F_l for the set of local regression functions $f_l \in F_l$ and require that all $f_l \in F_l$ are of the same type. The description length, or encoded size, of F is

$$L(F) = L_{\mathbb{N}}(|F|) + \log \binom{|X| - 1}{|F_l| - 1} + 2\log(|\mathcal{F}|) + L(f_g) + \sum_{f_l \in F_l} L(f_l),$$

where we first describe the number of local functions using $L_{\mathbb{N}}$, the MDL optimal encoding for integers $z \geq 1$ [28], then map each f_l to its associated value of X, after which we use $\log |\mathcal{F}|$ bits to identify the type of the global regression function f_g, and whenever F_l is non-empty also $\log |\mathcal{F}|$ bits to identify the type of the local regression functions f_l, finally, we encode the functions themselves. Knowing the type of a function, we only need to encode its parameters, and hence

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

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

Complexity of the conditional data

Reconstructing the data of Y given $f(X)$ corresponds to encoding the residuals, or the error the model makes. Since we fit our regression functions by minimizing the sum of squared errors, which corresponds to maximizing the likelihood under a Gaussian, it is a natural choice to encode the errors using a Gaussian distribution with zero-mean.

Since we have no assumption on the standard deviation, we use the empirical estimate $\hat{\sigma}$ to define the standard deviation of the Gaussian. By doing so, the encoded size of the error of $F(X)$ with respect to the data of Y corresponds to

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

where n_f is the number of data points for which we use a specific function $f \in F$. Intuitively, this score is higher the less structure of the data is described by the model and increases proportionally to the sum of squared errors.

Complexity of the conditional

Having defined the data and model costs above, we can now proceed and define the total encoded size of the conditional distribution of Y given X as

$$L(Y \mid X) = L(F) + L(Y \mid F, X). \tag{4}$$

By MDL, we are after that model F that minimizes Eq. (4). After discussing a significance test for our score, we will present the SLOPE algorithm to efficiently compute the conditional score in the next section.

4 Identifiability and significance

As it is not only important to find the causal direction, but also to provide some insight into when to trust the method and when to be careful, we here discuss identifiability and significance testing.

4.1 Identifiability

Determining for which generative processes and noise distributions we can reliably infer the causal direction from observational data alone is a non-trivial task; most of the work related to identifiability was done for additive noise models (ANMs) [9,23,30], and later generalized to Identifiable Functional Model Classes (IFMOCs) [24]. In the following, we discuss the relation of existing results on identifiability to SLOPE.

In causal inference based on ANMs, one assumes the generative model to be of the form $Y = f(X) + N$ where noise $N \perp\!\!\!\perp X$. If the data do admit such a model in the direction of $X \to Y$, but not in the opposite direction, we infer that X causes Y. As the functional form of the method is known, and the independence between residual and source measured are directly, it is relatively straightforward to determine whether the correct direction is identifiable for a function class and noise distribution [9,23,30,37]—for example, if f is linear and both X and N are Gaussian distributed, the causal direction is not identifiable as it is possible to find an ANM in both direction.

For functional causal models, Peters et al. [24] defined the Identifiable Functional Model Classes (IFMOC), which includes *bivariate* identifiable functions as *linear*, whereas *either* the cause or the independent noise can be *Gaussian* noise or *nonlinear* with both cause and noise being *Gaussian*. In essence, IFMOCs generalize individual results for additive noise models [9,23,30,37].

Janzing and Schölkopf [10] show that the statistic model of causality is closely linked to the more general algorithmic version. In particular, both the statistic and the algorithmic causal model assume that the local and global Markov conditions for the causal DAG G are fulfilled.

Local Markov condition: *For each node* $X_j \in G$, *it holds that* $X_j \perp\!\!\!\perp ND_J \mid PA_j$, *where* ND_j *are the non-descendants and* PA_j *the parents of* X_j.

Global Markov condition: *Given the sets of nodes* S, T, R. *If and only if* S *and* T *are d-separated by* R, *then* $S \perp\!\!\!\perp T \mid R$.

Further, both frameworks assume the same generative model. The functional model of causality and the corresponding algorithmic model of causality state that X_j can be generated as a function, respectively a program, of its parents and jointly independent noise [10]. Notably, a program can also be a functional relationship, which means that the algorithmic model of causality includes the functional model as well. Based on the previous statements, the authors postulate that the decomposition

$$K(P(X_1, \ldots, X_n)) \stackrel{+}{=} \sum_{j \in \{1,\ldots,n\}} K(P(X_j | PA_j))$$

only holds for the true causal DAG. In fact, Janzing and Steudel [11] show that it is unlikely that for both functional and the algorithmic causal models it is unlikely that in real-world data the additive noise assumption is violated.

Our inference rule is based on this postulate, and hence we inherit the property that the factorization of the joint in the true distribution is simpler in terms of Kolmogorov complexity. We are able to determine this direction if we use Kolmogorov complexity as a measure of complexity. However, as Kolmogorov complexity is not computable, and not even approximable up to arbitrary precision, but only approximable from above [15], it is impossible to make general statements about identifiability for any method build on this model; we have to rely on the quality of our MDL approximation. This means that we can only refer to the IFMOC definition under the assumption that the MDL score perfectly approximates the Kolmogorov complexity. Ideal MDL does have this property, but is not useable in practice. Using Refined MDL [8], we can make such statements relative to a model class, but such scores are only efficiently computable for a small number of model classes. Given enough samples, a two-part MDL score behaves like a Refined MDL score [8]. We use a two-part MDL score, which means that given enough samples we approximate the optimal MDL score for the class of functions we consider up to a constant that only depends on the model class we consider.

The dominant part of our score is the regression error. If we either ignore the model cost, or, equivalent, if we allow only models for the same complexity, we have a direct connection to the recent results of Blöbaum et al. [3]. In essence, they showed that if the true functional relationship is invertible, monotonically increasing and two times differentiable, we can identify the causal direction based on regression errors. From their results, it shows the approach is most reliable in a low noise setups. Relating this back, our approach is similar but more general: we can compare different functions types to each other, whereas Blöbaum et al. fit the same function type in both directions.

To conclude, given enough samples our approach is expected to be reliable for IFMOCs with the restriction that the function is in our model class and that we can model the noise distribution. In addition, we expect our approach to perform better in low noise setups. As a consequence, we can extend the class of identifiable functions by (i) extending the function class and (ii) by fitting functions that minimize a different error function. In addition, we can increase reliability by providing a significance value for an inference. For that manner, we propose two significance tests in the following subsections.

4.2 Significance by hypercompression

Ranking based on confidence works well in practice. Ideally, we would additionally like to know the significance of an inference. It turns out we can define an appropriate hypothesis test using the no-hypercompressibility inequality [4,8]. In a nutshell, under the hypothesis that the data were sampled from the null model, the probability that any other model can compress k bits better is

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

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

$$L_0 = \max\{L(X \to Y), L(Y \to X)\}.$$

The probability that the data can be compressed

$$k = |L(X \to Y) - L(Y \to X)|$$

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

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

$$L_0 = \min\{L(X \to Y), L(Y \to X)\} + |L(X \to Y) - L(Y \to X)|/2.$$

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

$$k = \frac{|L(X \to Y) - L(Y \to X)|}{2}.$$

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

4.3 Significance by confidence

Although the above significance test based on the absolute difference in compression follows nicely from theory, and behaves well in practice, it is not free of problems. In particular, as most significance tests, it is sensitive to the number of samples, which in our context can be directly linked to the initial complexities $L(X)$ and $L(Y)$. Assume X and X' follow the exact same distribution, where X contains 100 samples and X' 10,000. Further, Y, respectively Y', have been generated by the same process, as a function over X, respectively X', plus additive noise. It is easy to see that $L(X)$ will be much smaller than $L(X')$. Despite this difference, we would observe that the confidence value for both processes is similar because it considers the gain relative to the unconditioned costs. The absolute difference between $|L(X \to Y) - L(Y \to X)|$ and $|L(X' \to Y') - L(Y' \to X')|$ will likely be larger for the pair X' and Y', as it contains more samples. In essence, we assume that it is easier to gain k bits for large data sets than for smaller data sets. Following this assumption, the above significance test using the absolute difference is biased towards larger data sets.

To resolve this bias, we can reformulate the null hypothesis with respect to the marginal complexity. One way to do so is to rescale the initial complexity $L(X) + L(Y)$ to b bits. We write the new null hypothesis as H_0: *Given a budget of b bits, both directions compress equally well.* With this hypothesis, we calculate k as

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

This means that finding a threshold for the confidence value is equivalent to the relative significance test. In particular, we can calculate a confidence threshold given a significance level α and a budget b as $\mathbb{C} = -2\log(\alpha)/b$. For instance, allowing a budget of $b = 1000$ bits and a significance level of $\alpha = 0.05$ renders all inferences with a confidence value lower than 0.00864 insignificant. Informally, we say that we do not expect that a difference of more than k in b bits is due to a random effect.

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

Algorithm 1: CONDITIONALCOSTS(Y, X)

input : random variables Y and X
output : score $L(Y \mid X)$

1 $F =$ empty model;
2 $f_g =$ FITDETERMINISTIC($Y \sim X, \mathcal{F}$);
3 $F = F \cup f_g$;
4 $s = s_g = L(F) + L(Y \mid F, X)$;
5 $X_u = \{x \in X \mid \text{count}(x) \geq 2\}$;
6 **foreach** $\mathcal{F}_c \in \mathcal{F}$ **do**
7 $\quad s_c = s_g,\ F_c = F$;
8 $\quad$ **foreach** $x_i \in X_u$ **do**
9 $\quad\quad Y_i = \{y \in Y \mid y \text{ maps to } x_i\}$;
10 $\quad\quad X_i = \text{norm}(1 : |Y_i|, \min = -t, \max = t)$;
11 $\quad\quad f_l =$ FITDETERMINISTIC($Y_i \sim X_i, \mathcal{F}_c$);
12 $\quad\quad \hat{s} = L(F_c \cup f_l) + L(Y \mid F_c \cup f_l, X)$;
13 $\quad\quad$ **if** $\hat{s} < s_c$ **then** $s_c = \hat{s},\ F_c = F_c \cup f_l$
14 $\quad$ **if** $s_c < s$ **then** $s = s_c$
15 **return** s;

5 The SLOPE algorithm

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

5.1 Calculating the conditional scores

Algorithm 1 describes the subroutine to calculate the conditional costs $L(Y \mid X)$ or $L(X \mid Y)$. We start with fitting a global function f_g for each function class $c \in \mathcal{F}$ and choose the one f_g with the minimum sum of data and model costs (line 2). Next, we add f_g to the model F and store the total costs (3–4). For purely deterministic functions, we are done.

If X includes duplicate values, however, we need to check whether fitting a non-deterministic model leads to a gain in compression. To this end, we have to check for each value x_i of X that occurs at least twice, whether we can express the ascendingly ordered corresponding Y values, Y_i, as a function f_l of uniformly distributed data X_i between $[-t, t]$, where t is a user-determined scale parameter (lines 9–12). If the model costs of the new local function f_l are higher than the gain on the data side, we do not add f_l to our model (13). As it is fair to assume that for truly non-deterministic data the generating model for each local component is the same, we hence restrict all local functions to be of the same model class $\mathcal{F}_c \in \mathcal{F}$. As final result, we return the costs according to the model with the smallest total encoded size. In case of deterministic data, this will be the model containing only f_g.

5.2 Causal direction and confidence

In the previous paragraph, we described Algorithm 1, which is the main algorithmic part of SLOPE. Before applying it, we first normalize X and Y to be from the same domain and then determine the data resolutions τ_X and τ_Y for X and Y. To obtain the data resolution, we calculate the smallest nonzero difference between two instances of the corresponding random variable. Next, we apply Algorithm 1 for both directions to obtain $L(Y \mid X)$ and $L(X \mid Y)$. Subsequently, we estimate the marginals $L(X)$ and $L(Y)$ based on their data resolutions. This we do by modelling both as a uniform prior with $L(X) = -n \log \tau_X$ and $L(Y) = -n \log \tau_Y$. In the last step, we compute $\hat{\Delta}_{X \to Y}$ and $\hat{\Delta}_{Y \to X}$ and report the causal direction as well as the corresponding confidence value $\mathbb{C}$.

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

5.3 Combining basis functions

To extend the generality of SLOPE, we provide a second version of it, which we call SLOPER. The aim of SLOPER is to allow for more complex functions, e.g. $Y = a + bX + c \log(X) + dX^{-3} + N$. This we do by fitting a mixture of basis functions as the global function. As a consequence, SLOPER is more flexible and can help to infer more complex functional relationships. Naturally, this comes at a cost. In particular, we go over each possible combination of basis functions—in our case $2^{|\mathcal{F}|} - 1$ with $|\mathcal{F}| = 8$ basis functions—and find the one minimizing the two-part costs.

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

5.4 Computational complexity

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

6 Related work

Causal inference from observational data is an important open problem that has received a lot of attention in recent years [5,21,22,29]. Traditional constraint-based approaches, such as conditional independence test, require at least three random variables and cannot decide between Markov equivalent causal DAGs [22,33]. In this work, we focus specifically on those cases where we have to decide between the Markov equivalent DAGs $X \to Y$ and $Y \to X$.

A well-studied framework to infer the causal direction for the two-variable case relies on the additive noise assumption [30]. Simply put, it makes the strong assumption that Y is generated as a function of X plus additive noise, $Y = f(X) + N$, with $X \perp\!\!\!\perp N$. It can then be shown that while such a function is admissible in the causal direction, this is not possible in the anti-causal direction. There exist many approaches based on this framework that try to exploit linear [30] or nonlinear functions [9] and can be applied to real valued [9,25,30,37] as well as discrete data [23]. Recently, Mooij et al. [21] reviewed several ANM-based approaches from which ANM-pHSIC, a method employing the Hilbert–Schmidt Independence Criterion (HSIC) to test for independence, performed best. For ANMs, the confidence value is often expressed as the negative logarithm of the p value from the used independence test [21]. p values are, however, quite sensitive to the data size [1], which leads to a less reliable confidence value. As we will show in the experiments, our score is robust and nearly unaffected by the data size.

Another class of methods rely on the postulate that if $X \to Y$, the marginal distribution of the cause $P(X)$ and the conditional distribution of the effect given the cause $P(Y \mid X)$ are independent of each other. The same does not hold for the opposite direction [10]. The authors of IGCI define this independence via orthogonality in the information space. Practically, they define their score using the entropies of X and Y [12]. Liu and Chan implemented this framework by calculating the distance correlation for discrete data between $P(X)$ and $P(Y \mid X)$ [16]. A third approach based on this postulate is CURE [29]. Here, the main idea is to estimate the conditional using unsupervized inverse Gaussian process regression on the corresponding marginal and compare the result to the supervized estimation. If the supervized and unsupervized estimation for $P(X \mid Y)$ deviate less than those for $P(Y \mid X)$, an independence of $P(X \mid Y)$ and $P(X)$ is assumed and causal direction $X \to Y$ is inferred. Although well formulated in theory, the proposed framework is only solvable for data of up to 200 data points and otherwise relies strongly on finding a good sample of the data.

Recently, Janzing and Schölkopf postulated that if $X \to Y$, the complexity of the description of the joint distribution in terms of Kolmogorov complexity, $K(P(X, Y))$, will be shorter when first describing the distribution of the cause $K(P(X))$ and then describing the distribution of the effect given the cause $K(P(Y \mid X))$ than vice versa [10,14]. To the best of our knowledge, Mooij et al. [20] were the first to propose a practical instantiation of this framework based on the Minimum Message Length principle (MML) [35] using Bayesian priors. Vreeken [34] proposed to approximate the Kolmogorov complexity for numeric data using the cumulative residual entropy and gave an instantiation for multivariate continuous-valued data. Perhaps most related to SLOPE is ORIGO [5], which uses MDL to infer causal direction on binary data, whereas we focus on univariate numeric data.

7 Experiments

In this section, we empirically evaluate SLOPE and SLOPER. In particular, we consider synthetic data, a benchmark data set and a real-world case study. We implemented both in *R* and make both the code, the data generators and real-world data publicly available for research purposes.[1] We compare SLOPE and SLOPER to the state of the art for univariate causal inference. These include CURE [29], IGCI [12] and RESIT [25]. From the class of ANM-based methods, we compare to ANM-pHSIC [9,21], which a recent survey identified as the most reliable ANM inference method [21]. We use the implementations by the authors, sticking to the recommended parameter settings.

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

7.1 Evaluation measures

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

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

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

We use the accuracy curve and the area under the accuracy curve as our default measures and give additional results with respect to ROC and PR in the Appendix.

7.2 Synthetic data

We first consider data with known ground truth. To generate such data, we follow the standard scheme of Hoyer et al. [9]. That is, we first generate X randomly according to a given distribution and then generate Y as $Y = f(X) + N$, where f is a function that can be linear, cubic or reciprocal, and N is the noise term, which can either be additive or non-additive.

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

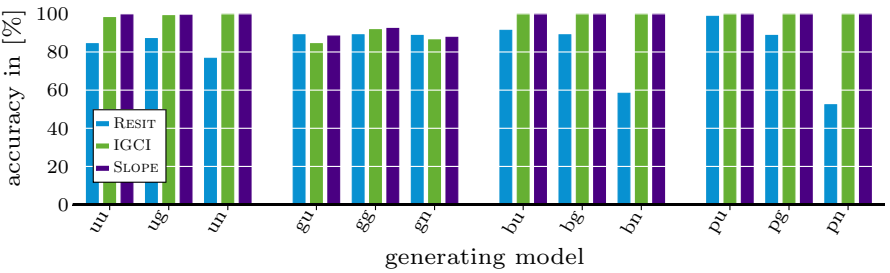


Fig. 2 [Higher is better] Accuracies of SLOPE, RESIT and IGCI on synthetic data (SLOPER performs identical to SLOPE). The first letter of the labels corresponds to the distribution of X (u uniform, g sub-Gaussian, b binomial and p Poisson), the second letter to that of the noise (u uniform, g Gaussian and n non-additive)

Accuracy

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

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

Note that the binomial and Poisson distribution generate discrete data points, which with high probability results in non-deterministic pairs. To generate Y, we first apply either a linear, cubic or reciprocal function on X, with fixed parameters, and add either additive noise using a uniform or Gaussian distribution with $t, s \sim \text{unif}(1, \max(x)/2)$ or non-additive noise with $\mathcal{N}(0, 1)|\sin(2\pi\nu X)| + \mathcal{N}(0, 1)|\sin(2\pi(10\nu)X)|/4$ according to [29], where we choose $\nu \sim \text{unif}(0.25, 1.1)$. For every combination, we generate 100 data sets of 1000 samples each.

Next, we apply SLOPE, RESIT and IGCI and record how many pairs they correctly infer. Since all tested functions can be modelled by SLOPE, they can also be modelled by SLOPER. Hence, the performance of SLOPER is identical, and we only give the results for one of them. As they take up to hours to process a single pair, we do not consider CURE and ANM here. We give the averaged results over all three function types in Fig. 2. In general, we find that SLOPE and IGCI perform on par and reach 100% for most setups, whereas SLOPE performs better on the sub-Gaussian data. If we consider the single results for linear, cubic and reciprocal, we find that on the linear data with sub-Gaussian distributed X, SLOPE performs on average 7% better than IGCI. We provide further details in terms of the non-aggregated results for only linear, cubic and reciprocal in the Appendix.

Confidence

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

[2] We consider sub-Gaussian distributions since linear functions with both X and N Gaussian distributed are not identifiable by ANMs [9].

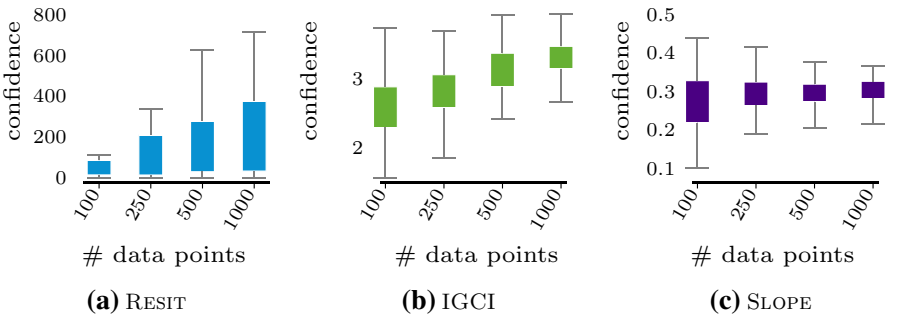


Fig. 3 [The more stable the better] Confidence values on a cubic function for different sample sizes. Unlike RESIT and IGCI, the SLOPE scores can be meaningfully compared between different sample sizes

To analyse this, we generate 100 data sets of 100, 250, 500 and 1000 samples each, where X is Gaussian distributed and Y is a cubic function of X with uniform noise. Subsequently, we apply RESIT, IGCI and SLOPE and record their confidence values. We show the results per sample size in Fig. 3. As each method uses a different score, the scale of the Y-axis is not important. What is important to note is the trend of the scores over different sample sizes. We see the mean of the confidence values of SLOPE is very consistent and nearly independent of the number of samples. In addition, our score becomes more precise with more data: the size of the confidence interval decreases. In strong contrast, the standard deviation of the confidence values increases with larger sample size for RESIT. For IGCI, we observe that the average confidence increases with the number of samples.

In addition to theses plots, we check whether there is a significant mean shift in the confidence values for different sample sizes. Hence, we apply the exact two-sided Wilcoxon rank-sum test [19,36]. In particular, we compare the confidence values for the sample sizes 100, 200, 500 to the ones for sample size 1000 for all methods. As a result, we observe that for a significance level of 0.01 we find a significant shift in all three tests for IGCI. Also, for RESIT, there is a significant mean shift between the values for 100 and 1000 as well as for 250 and 1000. SLOPE is consistent from 250 samples onwards. In other words, while it is easy to compare and rank SLOPE scores, this is not the case for the two others—which as we will see below results in comparatively bad accuracy curves.

7.3 Identifiability of ANMs on synthetic data

Connected to the vulnerability of p values, that RESIT uses, to the size of the data, we investigate in a similar problem. When the data size or the complexity of the function increases, the test for independence between X and N is likely to hold in both directions. Accordingly, we generate uniform data with Gaussian noise for different data sizes and plot the results for linear and cubic functions in Fig. 4. We can observe that this problem does very rarely occur for the linear data. For the more complex generative function, the cubic function, we observe that this problem frequently occurs. Notably, most of the time one direction is significant, the other is so, too. In such cases, RESIT and other ANM-based algorithms decide for the more extreme p value. As stated by Anderson et al. [1], a more extreme p value does not necessarily imply a stronger *independence*. The only valid statement we can make is that it is highly unlikely that the noise is *dependent* on X as well as on Y for the inverse direction. Deciding for the correct direction, however, is not well defined, especially if we consider

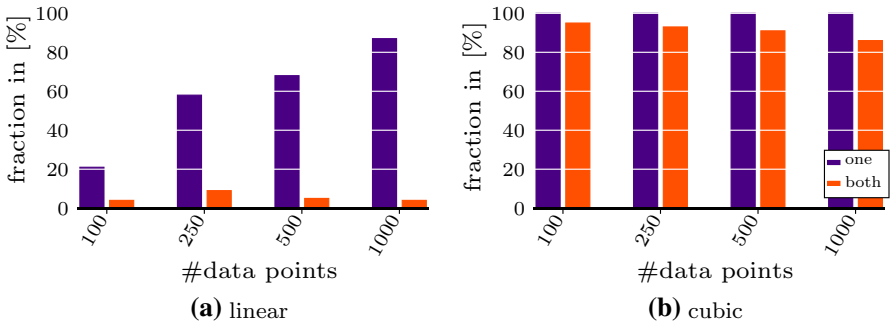


Fig. 4 Percentage of cases where one or both causal directions are significant under an ANM

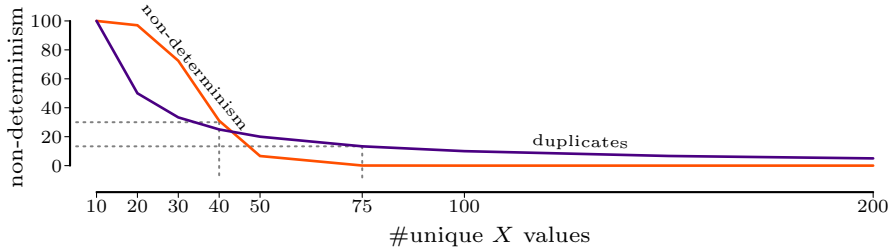


Fig. 5 [SLOPE does not overfit] Percentage of non-deterministic models SLOPE chooses, resp. the expected number of Y values per X value, for the number of unique values of X

that the p values can be very low and in the order of 10^{-100}, as we saw in the previous experiment.

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

Non-determinacy

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

To control non-determinacy, we sample X uniformly from k equidistant values over $[0, 1]$, i.e. $X \in \left[\frac{0}{k}, \frac{1}{k}, \ldots, \frac{k}{k}\right]$. To obtain Y, we apply a linear function and additive Gaussian noise as above. Per data set, we sample 1000 data points.

In Fig. 5, we plot the non-determinism of the model, i.e. the average number of used bins divided by the average number of bins SLOPE could have used, against the number of distinct X values. As a reference, we also include the average number of values of Y per value of X. We see that for at least 75 unique values, SLOPE does not infer non-deterministic models. Only at 40 distinct values, i.e. an average of 25 duplicates per X, SLOPE consistently starts to fit non-deterministic models. This shows that if anything, rather than being prone to overfit, SLOPE is conservative in using local models.

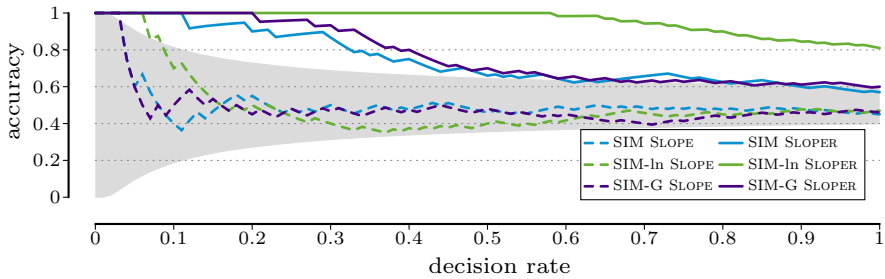


Fig. 6 [Higher is better] Accuracy curves of Slope and Sloper on the SIM, SIM-ln and SIM-G data sets. The grey area refers to the 95% confidence interval of a random coin flip

GP simulated data

Next, we want to show that considering a richer function class is beneficial for our approach. As a showcase, we apply both Slope and Sloper to the synthetic data pairs proposed by Mooij et al. [21], where both the data over the cause X and the function that maps X to Y have been generated using a Gaussian process. We consider three scenarios,[3] each containing 100 pairs of 1000 samples. The first one, *SIM* is the standard setup, *SIM-ln* has low noise levels and for *SIM-G* both the distribution of X and the additive noise are near Gaussian.

In Fig. 6 we provide the accuracy curves for Slope and Sloper. Overall, we can observe that Sloper clearly improves upon the results of Slope, since it is able to fit the more complex GP functions better. Especially for the low noise scenario, Sloper improves significantly and reaches an overall accuracy of 80%. In general, we can observe that the accuracy curves for both are good since the correct decisions have the highest confidence values.

If we consider the area under the accuracy curve, Sloper performs well having an AUAC of 96% on SIM-ln, 77% on SIM-G and 75% on SIM, whereas Slope has an AUAC of about 50% for all of them. As we expect our approach to work better in a low noise setup, it is not surprising that Sloper performs best on the SIM-ln data set.

7.4 Real-world data

Next, we evaluate Slope on real-world benchmark data. In particular, we consider the Tübingen cause–effect data set.[4] At the time of writing, the data set included 98 univariate numeric cause–effect pairs. We first compare Slope to IGCI, Resit, ANM and Cure, using their suggested parameter settings for this benchmark. Afterwards, we compare between different variants of Slope.

Accuracy curves and overall accuracy

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

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

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

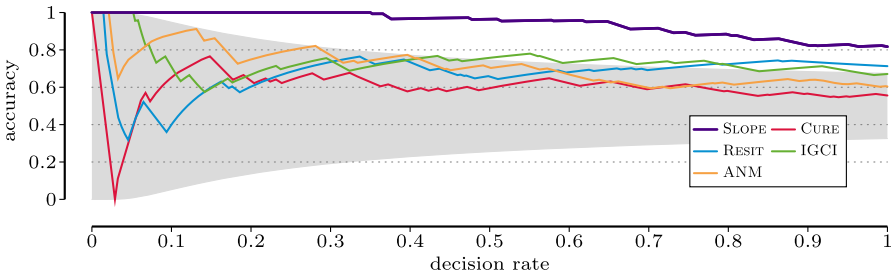


Fig. 7 [Higher is better] Accuracy curves of SLOPE, CURE, RESIT, IGCI and ANM on the Tübingen benchmark data set (98 pairs)

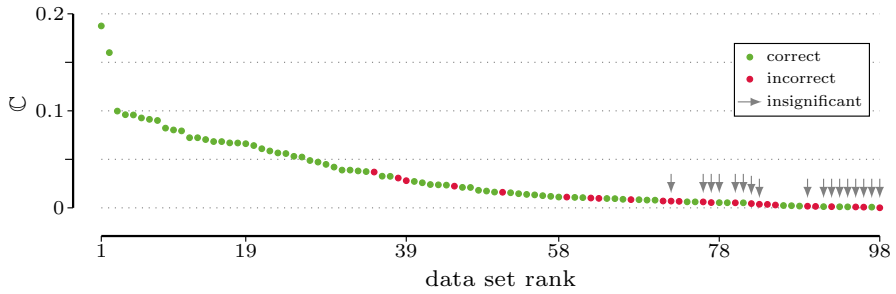
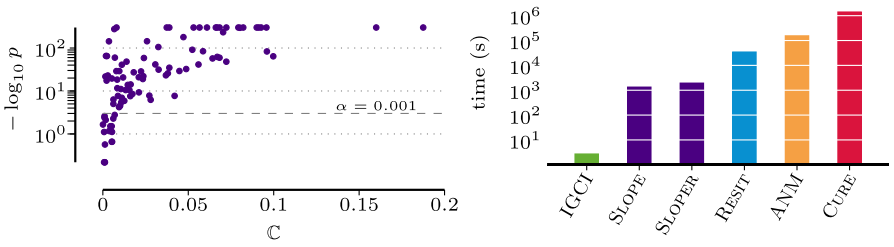


Fig. 8 Confidence values of SLOPE for the Tübingen benchmark pairs, in descending order, corresponding to Fig. 7. Correct inferences are marked in green, errors in red, and inferences insignificant at $\alpha = 0.001$ for the absolute p value test are marked with a grey arrow

We plot the results in Fig. 7, where in addition we show the 95% confidence interval for the binomial distribution with success probability of 0.5 in grey. We observe that SLOPE strongly outperforms its competitors in both area under the accuracy curve and overall accuracy; it identifies the correct result for top-ranked 34 data sets, over the top-72 pairs (which correspond to 72.4% of the weights) it has an accuracy of 90%, while over all pairs it obtains an accuracy of 81.7%.

In Fig. 8, we show the corresponding confidence values of SLOPE for the benchmark pairs. The plot emphasizes not only the predictive power of SLOPE, but also the strong correlation between confidence value and accuracy. In comparison with the other approaches, the area under the accuracy curve (Fig. 7) of SLOPE is stable and only decreases slightly at the very end. Our competitors obtain overall accuracies of between 56% (CURE) and 71% (RESIT), which for the most part are insignificant with regard to a fair coin flip. This is also reflected in the AUAC values, which lie between 0.588 (CURE) and 0.736 (IGCI), whereas SLOPE has an AUAC of 0.942.

If we not only consider the confidence values, but also our proposed statistical test based on the absolute difference, we can improve our results even further. After adjusting the p values using the Benjamini–Hochberg correction [2] to control the false discovery rate (FDR), 81 out of the 98 decisions are significant w.r.t. $\alpha = 0.001$. As shown in Fig. 8, the pairs rated as insignificant correspond to small confidence values. In addition, from the 17 insignificant pairs, 11 were inferred incorrect from SLOPE and 6 correct. Over the significant pairs, the weighted accuracy increases to 85.2%, and the AUAC to 0.965.



(a) $\mathbb{C}$ vs. p-value **(b)** runtime

Fig. 9 (Left) Confidence and significance of Slope on the Tübingen benchmark pairs. Only samples with low confidence are also insignificant. (Right) Runtime in seconds over all 98 pairs, in log-scale. Slope and Sloper both are more accurate than all, and faster than all except for IGCI

To provide further evidence that the confidence values and the p values are indeed related, we plot the adjusted p values and confidence values in Fig. 9a. We observe that high confidence values correspond to highly significant p values. We also computed the area under the accuracy curve for Slope when ranking by p values, and find it is only slightly worse than that ranked by confidence. We posit that confidence works better as it is more independent of the data size. To test this, we calculate the correlation between data size and corresponding measures using the maximal information coefficient (MIC) [26]. We find a medium correlation between confidence and p values (0.64), and between p values and data size (0.55), and only a weak correlation between confidence and data size (0.31).

Apart from the accuracies, we also tracked which functional dependencies Slope found on the benchmark data. We found that most of the time (54.6%), it fits linear functions. For 23.7% of the data, it fits exponential models, and for 15.5% cubic models. Quadratic and reciprocal models are rarely fitted (6.2%).

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

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

As explained in Sect. 4, the confidence value can be interpreted as a relative p value with respect to a given reference size, e.g. 1000 bits. Whereas ranking by the p value corresponding to the significance by confidence would obviously result in the same area under the accuracy curve as taking the confidence value itself, it does allow us to determine a sensible threshold to decide between significant and random decisions.

Given budget $b = 1000$ bits and a significance level $\alpha = 0.05$, we obtain a confidence threshold $\tau = 0.00864$. If we reconsider Fig. 8, we observe that 32 decisions are rendered insignificant by this threshold. From those, 17 are incorrect and 15 correct. Consequently, this threshold exactly prevents our algorithm to make 50 : 50, or random decisions. At the same time, considering only the significant decisions, results in an accuracy of 95.2%. Alternatively, if we lower the significance threshold to 0.01, eleven more decisions are insignificant, out of which more than two-thirds are correct, which implies that a $\alpha = 0.01$ might be too restrictive.

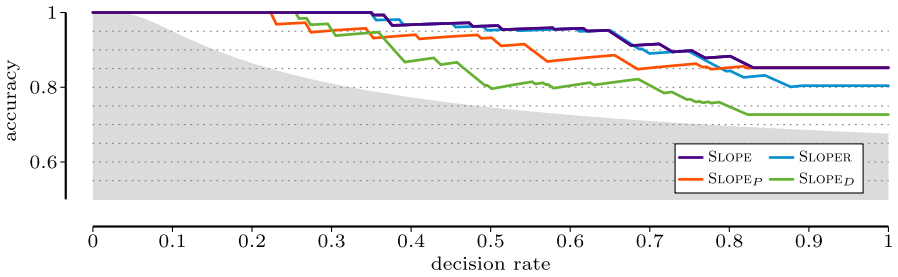


Fig. 10 [Higher is better] Accuracy curves for SLOPE, SLOPER, SLOPE$_P$ and SLOPE$_D$ on the Tübingen benchmark data set. SLOPE$_P$ is inferred with SLOPE, but ranked according to the p value and SLOPE$_D$ is an ablated version of SLOPE, which fits the data with a single deterministic function. Only significant decisions with respect to $\alpha = 0.001$ are considered

SLOPE *and its variations* As a last test on the benchmark data set, we compare SLOPE, SLOPER and two additional variants of our algorithm. Those are SLOPE$_D$, which only fits deterministic functions and SLOPE$_P$, which has the same results as SLOPE, but uses the absolute p value as confidence. For each variant, we plot the accuracy curve for all significant decisions with respect to the absolute significance test with $\alpha = 0.001$ in Fig. 10.

First of all, we observe that SLOPER is on par with SLOPE up to a decision rate of 75% and reaches an overall accuracy of 80%. The AUAC of SLOPER (0.936) is nearly as good as the one for SLOPE (0.945). Hence, only in the low confidence region, SLOPER had a slightly worse performance. When we inspected those decisions, we found that the corresponding pairs mainly consisted of pairs with high noise levels. This explains why SLOPE and SLOPER made different decisions as both were not very certain. Moreover, we observe that using the p value as confidence measure leads to a slightly worse accuracy curve and AUAC of 0.918; however, as expected it is still good as the confidence values correlate with the p values. SLOPE$_D$ has an overall accuracy of about 73% and an AUAC of 0.861, which clearly shows the necessity of fitting non-deterministic functions.

7.5 Runtime

Next, we evaluate the computational efficiency of SLOPE and SLOPER. To this end we report, per method, the wall-clock time needed to decide on all 98 pairs of the benchmark data set. We ran these experiments on Linux servers with two six-core Intel Xenon E5-2643v2 processors and 64GB RAM. The implementations of SLOPE, IGCI and RESIT are single-threaded, whereas ANM and CURE are implemented in Matlab and use the default number of threads. We give the results in Fig. 9. We see that IGCI is fastest, followed by SLOPE and SLOPER, taking 1 475 rsp. 1 936 seconds to process all pairs. The other competitors are all at least one order of magnitude slower. Taking 13 days, CURE has the longest runtime. The large gain in runtime of SLOPE compared to RESIT, ANM and CURE rises from the fact that those methods employ Gaussian process regression to fit the functions.

7.6 Case study: Octet binary semiconductors

To evaluate real-world performance, we conduct a case study on octet binary semiconductors [6,32]. In essence, the data set includes the 82 different materials one can form by taking

one each from two specific groups of atoms, and of which the resulting material either forms a rocksalt or zincblende crystal structure. The aim of current research is to predict, given a set of physical properties, the crystal structure of the material. A key component to distinguish between both forms is the energy difference δ_E between rocksalt and zincblende. At the time of writing, it is not known which combination of physical properties can be used to calculate δ_E; however, there exist candidate that are known to have some impact [6,7]. Since the data set contains very high quality measurements, it is well suited as a case study for our method.

In particular, form the set of physical properties, which also contains derived properties consisting of combinations or log transformations, we extracted the top 10 that had the highest association with δ_e [17]. The point is that we know that all of these properties somehow influence δ_E, but an exact formula to calculate δ_E is not known yet. After consulting the domain experts, we thus obtain 10 new cause effect pairs. For each of those pairs, we define δ_E as X and one of the top 10 features as Y. Since the energy difference is influenced by the features, we can assume that $Y \to X$ is the true causal direction for all pairs. For more detailed information to the data set, we refer to Ghiringhelli et al. [6]. We make these extracted cause–effect pairs available for research purposes.[5]

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

8 Discussion

The experiments clearly show that SLOPE works very well. It performs well in a wide range of settings, on both synthetic and real-world data. In particular, on the latter it outperforms the state of the art, obtaining highly stable accuracy curves and an overall accuracy of more than 10% better than the state of the art. Our case study showed it makes sensible decisions. Most importantly, SLOPE is simple and elegant. Its models are easy to interpret, it comes with a stable confidence score, a natural statistical test, and is computationally efficient.

The core idea of SLOPE is to decide for the causal direction by the simplest, best fitting regression function. To deal with non-deterministic data, we allow our model to additionally use local regression functions for non-deterministic values of X, which the experiments show leads to a large increase in performance. Importantly, we employ local regression within an MDL framework; without this, fitting local regressors would not make sense, as it would lead to strong overfitting. Moreover, we extend SLOPE to SLOPER, to also fit combinations of basis functions which helps us to pick up more complex functional relationships.

A key advantage of our MDL-based instantiation of the algorithmic Markov condition, compared to HSIC-based independence tests and IGCI, is that our score is not dependent on the size of the data. This makes it possible to meaningfully compare results among different tests; this is clearly reflected in the stable decision rates. Another advantage is that it allows us to define a natural statistical test based on compression rates, which allows us to avoid insignificant inferences. We further showed the link between significance and confidence by introducing a relative significance measure. Although this test is invariant of the size of the data, it has the drawback of introducing a new parameter. To get high confidence results, we recommend to use the relative significance measure with a budget of 1000 bits.

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

Although the performance of SLOPE is impressive, there is always room for improvement. For instance, it is possible to improve the search for local components by considering alternate re-distributions of X', apart from uniformly ascending values. This is not trivial, as there exist $n!$ possible orders, and it is not immediately clear how to efficiently optimize regression fit over this space. More obviously, there is room to expand the set of function classes that we use at the moment—kernel based, or Gaussian process-based regression are powerful methods that, at the expense of computation, will likely improve performance further.

In addition, the topic of identifiability is rarely studied for practical instantiations of the algorithmic model of causality. Most findings are only based on the Kolmogorov complexity, which makes it hard to make accurate statements over the identifiability. Obviously, having an optimal encoding w.r.t. the model class would help, but this is hard to achieve for numeric data.

For future work, we additionally aim to consider the detection of confounding variables—an open problem that we believe our information theoretic framework naturally lends itself to—as well as to extend SLOPE to multivariate and possibly mixed-type data. We are perhaps most enthusiastic about leveraging the high accuracy of SLOPE towards inferring causal networks from biological processes without the need of conditional independence tests.

9 Conclusion

We studied the problem of inferring the causal direction between two univariate numeric random variables X and Y. To model the causal dependencies, we proposed an MDL-based framework employing local and global regression. Further, we proposed SLOPE, an efficient linear-time algorithm, to instantiate this framework. Further, we extend SLOPE to consider combinations of basis functions which allows us to fit more complex functions. Moreover, we introduced 10 new cause effect pairs from a material science data set.

Empirical evaluations on synthetic and real-world data show that SLOPE reliably infers the correct causal direction with a high accuracy. On benchmark data, at 82%, accuracy SLOPE outperforms the state of the art by more than 10% and provides a more robust accuracy curve, while additionally also being computationally more efficient. In future research, we plan to consider detecting confounding, causal inference on multivariate setting, and use SLOPE to infer causal networks directly from data.

Acknowledgements Open access funding provided by Max Planck Society. The authors wish to thank Panagiotis Mandros and Mario Boley for help with the octet binary causal pairs. Alexander Marx 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.

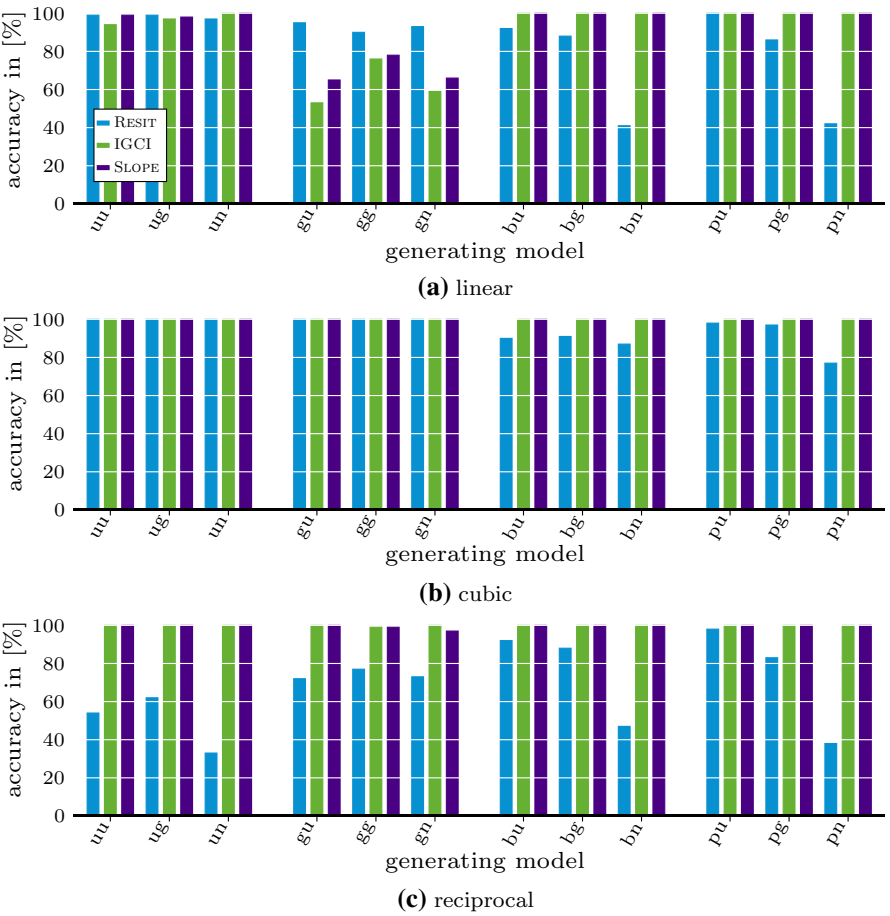


Fig. 11 [Higher is better] Accuracies of SLOPE, RESIT and IGCI on synthetic data (SLOPER has identical performance). The first letter of the labels corresponds to the distribution of X (u uniform, g sub-Gaussian, b binomial and p Poisson), the second letter to that of the noise (u uniform, g Gaussian and n non-additive)

Appendix

In this section, we give additional results and compare the area under the ROC, PR and accuracy curves as evaluation measures for causal inference.

Synthetic data For the synthetic data, we additionally provide the performance on only linear, cubic or reciprocal data in Fig. 11. We observe that SLOPE and IGCI are at $\sim 100\%$ accuracy for all scenarios except to sub-Gaussian distributed X with Y derived as a linear function. Nonetheless, SLOPE still reaches an accuracy of at least 65% on these data. IGCI, however, drops under 60% when Gaussian noise is added. RESIT on the other hand has a good performance on linear sub-Gaussian data but struggles with the reciprocal function as well as with linear binomial or Poisson data with Gaussian or nonlinear noise.

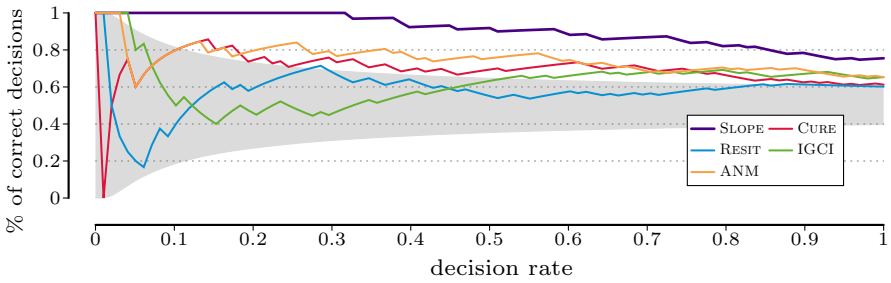


Fig. 12 [Higher is better] Accuracy curves for SLOPE, CURE, RESIT, IGCI and ANM on the *unweighted* Tübingen benchmark data set—i.e. the weights of all 98 pairs were set to one

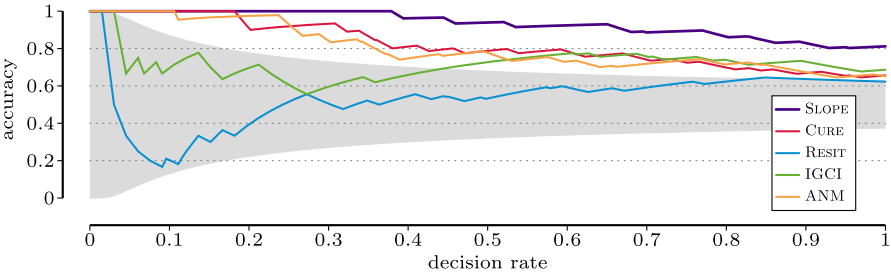


Fig. 13 [Higher is better] Weighted accuracy curves of SLOPE, CURE, RESIT, IGCI and ANM on the 0.9 version of the Tübingen benchmark data set (79 pairs)

Unweighted accuracy curves In this paragraph, we discuss the accuracy curves for the Tübingen benchmark data set, whereas we weight each decision equally—assigning it a weight of one. For the original plot, we weighted the results as recommended. The weights have the effect that multiple experiments on the same data set, which are very similar, get lower weights—e.g. their weights sum up to one full experiment. To show that the weighing is not just coincidentally in our favour, we give the results of the same experiment with every pair the same weight in Fig. 12. Although the overall accuracy of SLOPE decreases slightly to 75.5%, we still significantly outperform ANM, which is ranked on the second place with 66.0% accuracy.

Benchmark version 0.9 For those readers that are familiar with the competing approaches, we give one additional plot. The original papers for CURE [29], ANM [25], etc., report higher accuracies than we found above as they consider an older version of the benchmark data set that contains only 79 pairs, most of which with weight one. To allow for fair comparison on this setting, we give the results on this data in Fig. 13. CURE and ANM perform better on this subset, especially on the first few pairs. SLOPE, however, has roughly the same performance (81.2%) as for the whole data set. Quite a margin behind SLOPE is IGCI reaching 68.7% accuracy, followed by CURE, having an overall accuracy of 65.7%. Notably, although CURE here performs better than above, it does not quite reach the performance of 75% reported by the authors. One reason could be the probabilistic nature of the method, but otherwise we ran the experiments with the original code and the recommended parameter settings and therefore cannot fully explain the large difference in performance.

Table 1 [Higher is better] Area under the ROC, PR and accuracy curves for SLOPE, SLOPER, CURE, RESIT, IGCI and ANM on both the Tübingen data set including 98 univariate pairs and the older version 0.9, including only 79 univariate pairs. All decisions are weighted with the corresponding weights of the benchmark

	SLOPE	SLOPER	CURE	RESIT	IGCI	ANM
Tübingen_{98}						
ROC_X	**0.898**	0.865	0.424	0.573	0.671	0.472
ROC_Y	**0.897**	0.862	0.413	0.564	0.675	0.472
PR_X	**0.962**	0.948	0.716	0.791	0.808	0.734
PR_Y	**0.728**	0.705	0.232	0.265	0.600	0.255
AUAC	**0.942**	0.927	0.588	0.676	0.736	0.713
Tübingen_{79}						
ROC_X	**0.812**	0.792	0.381	0.508	0.388	0.469
ROC_Y	**0.851**	0.830	0.414	0.528	0.422	0.502
PR_X	**0.942**	0.935	0.740	0.800	0.675	0.742
PR_Y	**0.575**	0.573	0.200	0.254	0.269	0.232
AUAC	**0.933**	0.924	0.819	0.534	0.715	0.802

Bold values highlighted the best scoring

Area under the ROC, PR and accuracy curve Next, we briefly discuss different evaluation measures as the area under the ROC, PR and accuracy curve. We use each measure to evaluate SLOPE, SLOPER, CURE, RESIT, IGCI and ANM on both the Tübingen data set including 98 univariate pairs and the older version 0.9, including only 79 univariate pairs. For the ROC and PR curves, we compute both directions, where ROC_X corresponds to selecting $X \to Y$ as true positive and ROC_Y to selecting $Y \to X$ as true positive—accordingly so for PR. We show the results in Table 1.

First of all, we observe that both SLOPE and SLOPER have very similar results and outperform the competing approaches on each scoring metric. In general, the results relate to the corresponding accuracy curves. Further, we observe that it makes a huge difference for every approach whether we consider PR_X or PR_Y. This difference relates to the imbalance of the benchmark data set. However, since it is an arbitrary choice how to assign X and Y for each data set, we consider the area under the decision recall curve not as an appropriate measure for causal inference. We observe a similar effect for the area under the ROC curve, but much weaker. Still the score is dependent on the choice of the true positive and hence we consider the area under the accuracy curve as the most objective measure.

References

1. Anderson D, Burnham K, Thompson W (2000) Null hypothesis testing: problems, prevalence, and an alternative. J Wildl Manag 64(4):912–923
2. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B (Stat Methodol) 57(1):289–300
3. Blöbaum P, Janzing D, Washio T, Shimizu S, Schölkopf B (2018) Cause-effect inference by comparing regression errors. In: Proceedings of the international conference on artificial intelligence and statistics (AISTATS)
4. Bloem P, de Rooij S (2017) Large-scale network motif learning with compression. CoRR arXiv:1701.02026
5. Budhathoki K, Vreeken J (2017) Origo: causal inference by compression. Knowl Inf Syst 56:285–307
6. Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M (2015) Big data of materials science: critical role of the descriptor. Phys Rev Lett 114(10):1–5

7. Goldsmith BR, Boley M, Vreeken J, Scheffler M, Ghiringhelli LM (2017) Uncovering structure–property relationships of materials by subgroup discovery. New J Phys 19(1):013,031
8. Grünwald P (2007) The minimum description length principle. MIT Press, Cambridge
9. 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), Vancouver, BC, pp 689–696
10. Janzing D, Schölkopf B (2010) Causal inference using the algorithmic Markov condition. IEEE Trans Inf Technol 56(10):5168–5194
11. Janzing D, Steudel B (2010) Justifying additive noise model-based causal discovery via algorithmic information theory. Open Syst Inf Dyn 17(2):189–212
12. Janzing D, Mooij J, Zhang K, Lemeire J, Zscheischler J, Daniušis P, Steudel B, Schölkopf B (2012) Information-geometric approach to inferring causal directions. Artif Intell 182–183:1–31
13. Kolmogorov A (1965) Three approaches to the quantitative definition of information. Problemy Peredachi Informatsii 1(1):3–11
14. Lemeire J, Dirkx E (2006) Causal models as minimal descriptions of multivariate systems. http://parallel.vub.ac.be/index.htm?http://parallel.vub.ac.be/~jan/ (unpublished data)
15. Li M, Vitányi P (1993) An introduction to Kolmogorov complexity and its applications. Springer, Berlin
16. Liu F, Chan L (2016) Causal inference on discrete data via estimating distance correlations. Neural Comput 28(5):801–814
17. Mandros P, Boley M, Vreeken J (2017) Discovering reliable approximate functional dependencies. In: Proceedings of the 23nd ACM international conference on knowledge discovery and data mining (SIGKDD), Halifax, Canada. ACM, pp 355–364
18. Marx A, Vreeken J (2017) Telling cause from effect using MDL-based local and global regression. In: Proceedings of the 17th IEEE international conference on data mining (ICDM), New Orleans, LA. IEEE, pp 307–316
19. Marx A, Backes C, Meese E, Lenhof HP, Keller A (2016) EDISON-WMW: exact dynamic programming solution of the Wilcoxon–Mann–Whitney test. Genomics Proteomics Bioinform 14:55–61
20. Mooij J, 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 (26), pp 1–9
21. Mooij JM, Peters J, Janzing D, Zscheischler J, Schölkopf B (2016) Distinguishing cause from effect using observational data: methods and benchmarks. J Mach Learn Res 17(32):1–102
22. Pearl J (2009) Causality: models, reasoning and inference, 2nd edn. Cambridge University Press, New York
23. 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), JMLR, pp 597–604
24. Peters J, Mooij J, Janzing D, Schölkopf B (2012) Identifiability of causal graphs using functional models. ArXiv preprint arXiv:1202.3757
25. Peters J, Mooij J, Janzing D, Schölkopf B (2014) Causal discovery with continuous additive noise models. J Mach Learn Res 15:2009–2053
26. Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC (2011) Detecting novel associations in large data sets. Science 334(6062):1518–1524
27. Rissanen J (1978) Modeling by shortest data description. Automatica 14(1):465–471
28. Rissanen J (1983) A universal prior for integers and estimation by minimum description length. Ann Stat 11(2):416–431
29. Sgouritsa E, Janzing D, Hennig P, Schoelkopf B (2015) Inference of cause and effect with unsupervised inverse regression. In: Proceedings of the international conference on artificial intelligence and statistics (AISTATS), vol 38, pp 847–855
30. Shimizu S, Hoyer PO, Hyvärinen A, Kerminen A (2006) A linear non-Gaussian acyclic model for causal discovery. J Mach Learn Res 7:2003–2030
31. Spirtes P, Glymour C, Scheines R (2000) Causation, prediction, and search. MIT Press, Cambridge
32. Van Vechten JA (1969) Quantum dielectric theory of electronegativity in covalent systems. I. Electronic dielectric constant. Phys Rev 182(3):891
33. 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
34. Vreeken J (2015) Causal inference by direction of information. In: Proceedings of the SIAM international conference on data mining (SDM), Vancouver, Canada. SIAM, pp 909–917
35. Wallace CS, Boulton DM (1968) An information measure for classification. Comput J 11(1):185–194
36. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83

37. 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). AUAI Press, pp 647–655

Alexander Marx is a Ph.D. student at the Max Planck Institute for Informatics and Saarland University. He holds a Ph.D. Fellowship from the International Max Planck Research School for Computer Science (IMPRS-CS). His research interests include scalable and robust methods for causal inference and discovery.

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. At the time of writing, he authored over 75 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.