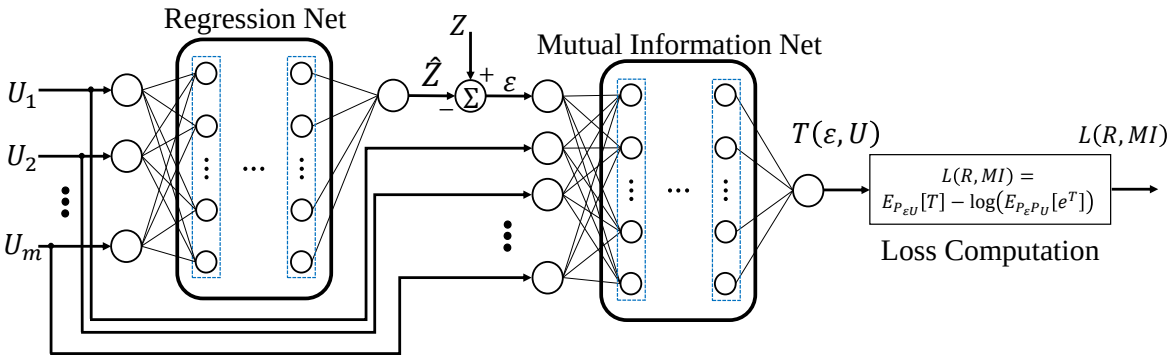
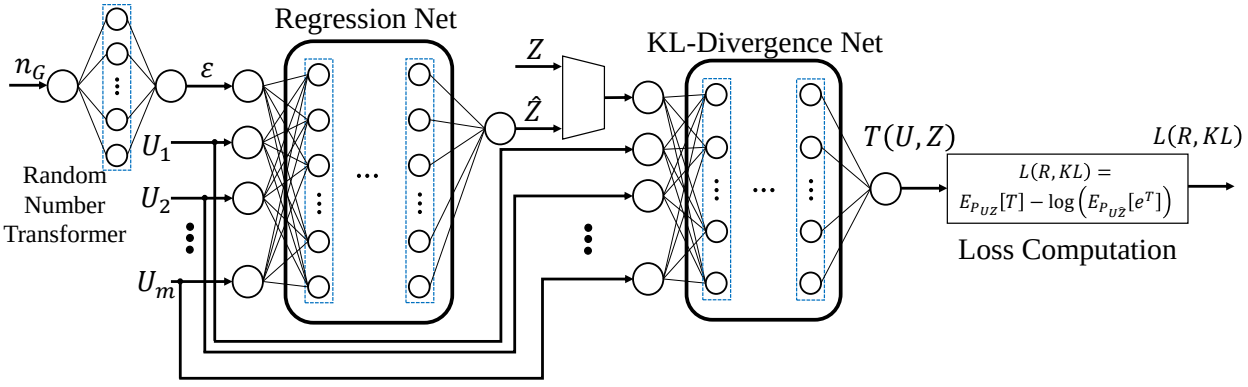
Graphical Abstract

Adversarial Orthogonal Regression: Two non-Linear Regressions for Causal Inference

M. Reza Heydari,Saber Salehkaleybar,Kun Zhang



a. AdOR Structure



b. AdOSE Structure

Adversarial Orthogonal Regression: Two non-Linear Regressions for Causal Inference

M. Reza Heydari[a], Saber Salehkaleybar[a,*] and Kun Zhang[b]

[a]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[b]Department of Philosophy, Carnegie Mellon University, Pittsburgh, United States

ARTICLE INFO

Keywords:
Orthogonal regression
Adversarial models
Additive noise model
Structural equation model
Mutual information

ABSTRACT

We propose two nonlinear regression methods, namely, Adversarial Orthogonal Regression (AdOR) for additive noise models and Adversarial Orthogonal Structural Equation Model (AdOSE) for the general case of structural equation models. Both methods try to make the residual of regression independent from regressors, while putting no assumption on noise distribution. In both methods, two adversarial networks are trained simultaneously where a regression network outputs predictions and a loss network that estimates mutual information (in AdOR) and KL-divergence (in AdOSE). These methods can be formulated as a minimax two-player game; at equilibrium, AdOR finds a deterministic map between inputs and output and estimates mutual information between residual and inputs, while AdOSE estimates a conditional probability distribution of output given inputs. The proposed methods can be used as subroutines to address several learning problems in causality, such as causal direction determination (or more generally, causal structure learning) and causal model estimation. Experimental results on both synthetic and real-world data demonstrate that the proposed methods have remarkable performance with respect to previous solutions.

1. Introduction

Identifying cause-effect relationships between variables in complex high dimensional networks has been studied in many fields such as neuroscience [15, 31], computational genomics [19, 10], economics [41], and social networks [36, 37]. For instance, in genomics, it is known that each cell of living creatures consists of a huge number of genes that produce proteins in a procedure called "gene expression," in which they can inhibit or promote each others' activities. These cause-effect relationships can be represented by a causal graph in which each variable is depicted by a node, and a directed edge that shows the direct causal effect from the "parent" node to the "child" node. It is commonly assumed that there is no directed cycle in the causal graph, i.e., it is a Directed Acyclic Graph (DAG). The goal is to recover the causal graph from the data sampled from variables. In the literature, learning causal graphs has been studied extensively in two main settings: random variables and time series.

In the setting of random variables, Shimizu et al. [32] proposed LiNGAM algorithm which can identify the causal graph in linear model under the assumption of non-Gaussianity of exogenous noises in the system. Hoyer et al. [12] proposed a method to reveal the direction of causality in additive noise model where the effect is a function of direct causes plus some exogenous noise. The basic idea of their method is the following: for a given candidate DAG, one solves a regression problem for each node, modeling it as a (possibly nonlinear) function of its parents. Then, a statistical independence test is performed to assess whether all residuals are jointly independent. If that is the case, the candidate DAG is accepted, otherwise it is rejected. Peters et al. [26] extended this idea for time series in additive noise models. All these methods require nonparametric nonlinear regression in order to check whether the residual is independent of regressors.

In the setting of time series, much efforts exerted to define statistical definition of causality such as Granger causality [8, 9]. Marko [20] defined an information theoretic measure called Directed Information (DI), which is a statistical criterion to detect the existence of direct causal effect between any pair of time series. Based on DI and inspired by Granger causality, Quinn et al. [29] showed that minimal generative model, i.e., a graph with minimum number of edges that does not miss the full dynamics, can be discovered by causally conditioned DI. Moreover, experiments showed that the proposed criterion can be used to reconstruct efficiently the causal graphs with linear relationships.

Causally conditioned DI and the other information theoretic measures for causality in time series typically utilize "differential entropy" [27], which is an extension of Shannon entropy for continuous random variables. Since differential entropy is defined based on the probability distribution, numerous works have been done for entropy estimation of general distributions using only observational data. In this regard, Hausser and Strimmer [11] used a naive binning method to estimate the value of joint distribution in each bin and then adjusted these values by a shrinkage factor based on James-Stein estimator [13]. In [18, 4, 21], the joint distribution is estimated by partitioning the domain in such a way that more accurate values are achieved in the regions where

*Corresponding author
heydari_mr@ee.sharif.edu (M. Reza Heydari); saleh@sharif.edu (S. Salehkaleybar); kunz1@cmu.edu (K. Zhang)
http://sina.sharif.ir/~saleh/ (S. Salehkaleybar); http://www.andrew.cmu.edu/user/kunz1/ (K. Zhang)
ORCID(s): 0000-0001-6862-0423 (M. Reza Heydari); 0000-0003-3934-9931 (S. Salehkaleybar)

the density of sampled data is high. However, the proposed methods are sophisticated and need huge computational cost in high dimension.

Recently, Quinn et al. [29] used a regression based method for estimating DI. In order to check whether a variable Y is the parent of variable X, two regressions are performed: one by considering the Y in the regressors, and another without it. Then, DI can be obtained by differing the entropy of residuals in two regressions. The variable Y is considered as a parent of X if DI is non-zero. The above procedure works correctly only if the obtained residuals are independent of regressors in both regressions.

According to what mentioned above, several causal learning algorithms in the setting of random variables (such as the one in [12]) or time series (such as TiMINo algorithm in [26], Granger causality analysis discussed in [42], and DI estimator in [29]), require a subroutine that can perform nonlinear regression such that the residual becomes independent of the regressors as much as possible. Basically, two principal assumptions are held in most of the regression problems: first, the residual is an additive term in problem formulation; secondly, its distribution is assumed to be Gaussian. Furthermore, common regression methods are confined to minimize Mean Squared Error (MSE) loss [38, 16]. Thus, in these common methods, the residuals and regressors become only uncorrelated. While these methods are fully efficient in linear Gaussian case, they might not be statistically efficient in nonlinear or non-Gaussian scenarios. To resolve this issue, Mooij et al. [22] proposed a novel regression method which minimizes the dependence between residuals and regressors that is meausured by Hilbert-Schmidt Independence Criterion (HSIC). In the proposed method, it is needed to carefully tune the kernel parameter in HSIC.

In this paper, we propose two non-parametric methods to circumvent such technical assumptions. In particular, our proposed methods take the advantage of adversarial training scheme, which was first proposed by Goodfellow et al. [7], and further works have exhibited the capacity of this framework to model underlying unknown distributions, especially in image generation task. Thanks to the adversarial training framework, the first method handles cases with additive exogenous noise, and the second method relaxes this assumption so it can be used in a more general setting.

Contributions: In this paper, we propose two nonlinear regression methods, namely, Adversarial Orthogonal Regression (AdOR) and Adversarial Orthogonal Structural Equation Model (AdOSE). AdOR assumes that the noise is modeled as an additive term while AdOSE relaxes this assumption. The models are "Adversarial", in the sense that in both methods, two neural networks compete with each other, the regression network and the loss network. In AdOR, the loss network estimates the mutual information between regressors and residuals, and in AdOSE, it acts as a Kullback-Leibler (KL)-divergence estimator between correct responses and predicts (which are the output of regression network). As discussed above, independence of residuals and regressors is vital in inferring the correct causal relationships. Thus, AdOR tries to make the residual independent of regressors, and AdOSE achieves this target by independently generating noise. The proposed methods can be used as subroutines to address several learning problems in causality, such as determining causal direction, causal structure learning, or causal model estimation. Experimental results show that the proposed methods have remarkable performance in estimating the true non-linear function with respect to previous solutions. While our main contribution is in causal inference, the proposed methods might also be useful in the other regression tasks.

The rest of the paper is organized as follows: In Section 2, we describe a neural network [2] that has been proposed previously to estimate mutual information. We present AdOR and AdOSE methods in Section 3 and Section 4, respectively. We provide experimental results in Section 5 and conculde the paper in Section 6.

2. Mutual Information Neural Estimation

In this section, we describe the neural network proposed in [2] for estimating mutual information based on an alternative representations of KL-divergence. This representation will be exerted as the loss network in Section 3 and 4.

Let P and Q be two distributions on some compact domain $\Omega \subset \mathbb{R}^d$. The KL-divergence between them is defined as:

$$D_{KL}(P||Q) := \mathbb{E}_P\left[\log\frac{dP}{dQ}\right]. \tag{1}$$

One of the representation of KL-divergence, which we focused on, is Donsker-Varadhan representation [5]:

$$D_{KL}(P||Q) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_P[T] - \log\left(\mathbb{E}_Q\left[e^T\right]\right), \tag{2}$$

where the supremum is taken over all functions T such that the two expectations are finite.

Let X and Y denote two continuous random variables with distributions P_X and P_Y, respectively. Mutual information between X and Y is denoted by $I(X;Y)$, which is a measure for the dependence of them. Mutual information has some multiple forms, and one form is defined as the KL-divergence between the joint distribution P_{XY}, and the product of marginal distributions $P_X P_Y$:

$$I(X;Y) = D_{KL}\left(P_{XY}\middle||P_X P_Y\right). \tag{3}$$

Let $\mathcal{F} = \left\{T_\theta\right\}_{\theta\in\Theta}$ be the set of functions parametrized by a neural network (i.e. weights, biases, batch normalization parameters, etc.). Mutual Information Neural Estimator (MINE, see definition 3.1 of [2]) is defined as:

$$\hat{I}(X;Y) = \sup_{\theta\in\Theta} \mathbb{E}_{P_{XY}}\left[T_\theta\right] - \log\left(\mathbb{E}_{P_X P_Y}\left[e^{T_\theta}\right]\right). \tag{4}$$

As the class of all functions in (2) is restricted to neural network class $\mathcal{F}$ in (4), we have the following lower bound:

$$I(X;Y) \geq \hat{I}(X;Y). \tag{5}$$

Theoretical properties of $\hat{I}(X;Y)$ are provided in [2]. In MINE, samples from joint distribution P_{XY} are fed as the inputs of a neural network and an optimizer like stochastic gradient descent, updates the parameters θ so as to maximize the right hand side of (4). Ultimately, as the parameters converge, the loss value of network is the estimated mutual information. For more details on the implementation of MINE, please refer to Algorithm 1 of [2].

Estimator (4) is the basis of idea behind AdOR. It is used as an unbiased consistent mutual information estimator to measure high order dependency of two random variables. This estimator is more accurate than the other estimators even for high dimensional variables as shown in the experiment part of [2]. Note that we use the first representation of KL-divergence (2) for the case of AdOSE in Section 4.

3. Adversarial Orthogonal Regression

Assume that U represents regressor vector, and Z is scalar dependent variable with the following unknown relationship:

$$Z = f(U) + \epsilon, \tag{6}$$

where ϵ is the residual term. We do not put any assumption on distributions of U, ϵ or the shape of $f(.)$. The regression problem is to find $\hat{f}$:

$$\hat{Z} = \hat{f}(U), \tag{7}$$

such that the residual $\varepsilon = Z - \hat{Z}$ is independent of U.

In AdOR method, the regression network (R) is pitted against the loss network where a mutual information estimator (MI) learns to find any high order dependencies (see the top block diagram of Figure 1). In the regression part, $\hat{Z} = \hat{f}(U;\theta_R)$ is a differentiable function represented by a multilayer perceptron, and parametrized with θ_R, in which $\hat{Z}$ is the regression output. The residual $\varepsilon = Z - \hat{Z}$ and the regressor vector U are fed as inputs to MI, and the output $T(\varepsilon, U; \theta_{MI})$ is also a differentiable function represented by a multilayer perceptron with parameters θ_{MI}. $L(R, MI) = \mathbb{E}_{P_{\varepsilon U}}[T] - \log\left(\mathbb{E}_{P_\varepsilon P_U}\left[e^T\right]\right)$ denotes the mutual information between U and ε. R is trained to minimize the dependency between residual and regressors. MI is simultaneously trained to tighten the gap between $I(U;\varepsilon)$ and $\hat{I}(U;\varepsilon)$ in order to achieve more accurate estimate of mutual information. In other words, R and MI play the following two-player minimax game:

$$\min_R \max_{MI} L(R, MI) = \mathbb{E}_{P_{\varepsilon U}}[T] - \log\left(\mathbb{E}_{P_\varepsilon P_U}\left[e^T\right]\right). \tag{8}$$

At equilibrium point, the value of loss $L(R, MI)$ is mutual information between U and ε. We provide experimental results in Section 5.1 that show convergence to the equilibrium point. In practice, the game in (8) is implemented by an iterative approach, in which the gradient of loss ∇L_B for mini-batch B is used via back-propagation procedure. As mentioned in [2], the denominator of the second term in the mini-batch's gradient $\nabla L_B = \mathbb{E}_B[\nabla T] - \frac{\mathbb{E}_B[\nabla T e^T]}{\mathbb{E}_B[e^T]}$ leads to a biased estimate of the full-batch gradient ∇L. To overcome this issue, Belghazi et al. [2] proposed to replace the estimator in denominator by an exponential moving average. This strategy reduces the bias term for small learning rates. In contrast, Adam optimizer [17] can be utilized where the history of gradients is also considered in the next update to reduce the bias term. Furthermore, the adjustable learning rate accelerates the procedure at early iterations, while the bias term declines as the model reaches to the optimal point.

Algorithm 1: AdOR

for *number of iterations* **do**

Forward path:

1. Draw $2b$ minibatch samples $\left\{\left(u^{(1)}, z^{(1)}\right), \ldots, \left(u^{(2b)}, z^{(2b)}\right)\right\}$
2. Evaluate regression output $\hat{z}^{(i)} = \hat{f}\left(u^{(i)}; \theta_R\right); i = 1, \ldots, 2b$
3. Compute residual $\varepsilon^{(i)} = z^{(i)} - \hat{z}^{(i)}; i = 1, \ldots, 2b$
4. Evaluate output of MI twice $T^{(i)} = T\left(\varepsilon^{(i)}, u^{(i)}; \theta_{MI}\right); i = 1, \ldots, b$ $T^{(i)}_{sh} = T\left(\varepsilon^{(i+b)}, u^{(i)}; \theta_{MI}\right); i = 1, \ldots, b$
5. Compute loss $L_B\left(\theta_R, \theta_{MI}\right) = \frac{1}{b}\sum_{i=1}^{b} T^{(i)} - \log\left(\frac{1}{b}\sum_{i=1}^{b} e^{T^{(i)}_{sh}}\right)$

Backward path:

for k_R *steps* **do**
  Update R by descending its stochastic gradient $\nabla_{\theta_R} L_B$
end

for k_{MI} *steps* **do**
  Update MI by ascending its stochastic gradient $\nabla_{\theta_{MI}} L_B$
end

end

Algorithm 1 shows the training phase of AdOR. In forward path, $2b$ examples are fed to R, and residuals $\varepsilon^{(i)}$ are computed in line 3. The first b pairs $\varepsilon^{(i)}$ and $u^{(i)}$ are joint samples; while, the second b pairs $\varepsilon^{(i+b)}$ and $u^{(i)}$ are marginal samples. Output of MI is computed twice: once by joint samples, and once by marginal samples in line 4. Finally, mini-batch loss L_B is computed in line 5 based on mean of samples computed in line 4. In backward path, parameters of each network are updated while the ones of other network is fixed. Note that in each iteration, R and MI are updated k_R and k_{MI} times, respectively.

We assume that the network's structure is fully connected for simplicity. But, one can use other structures like convolutional, recurrent, or any preferred structures. Note that the second b samples are drawn in each iteration to estimate marginal expectation in (4), which is empirically computed in line 4 of algorithm 1. After that the model is trained, there is no more need to the second b samples for finding regression output $\hat{f}(U)$.

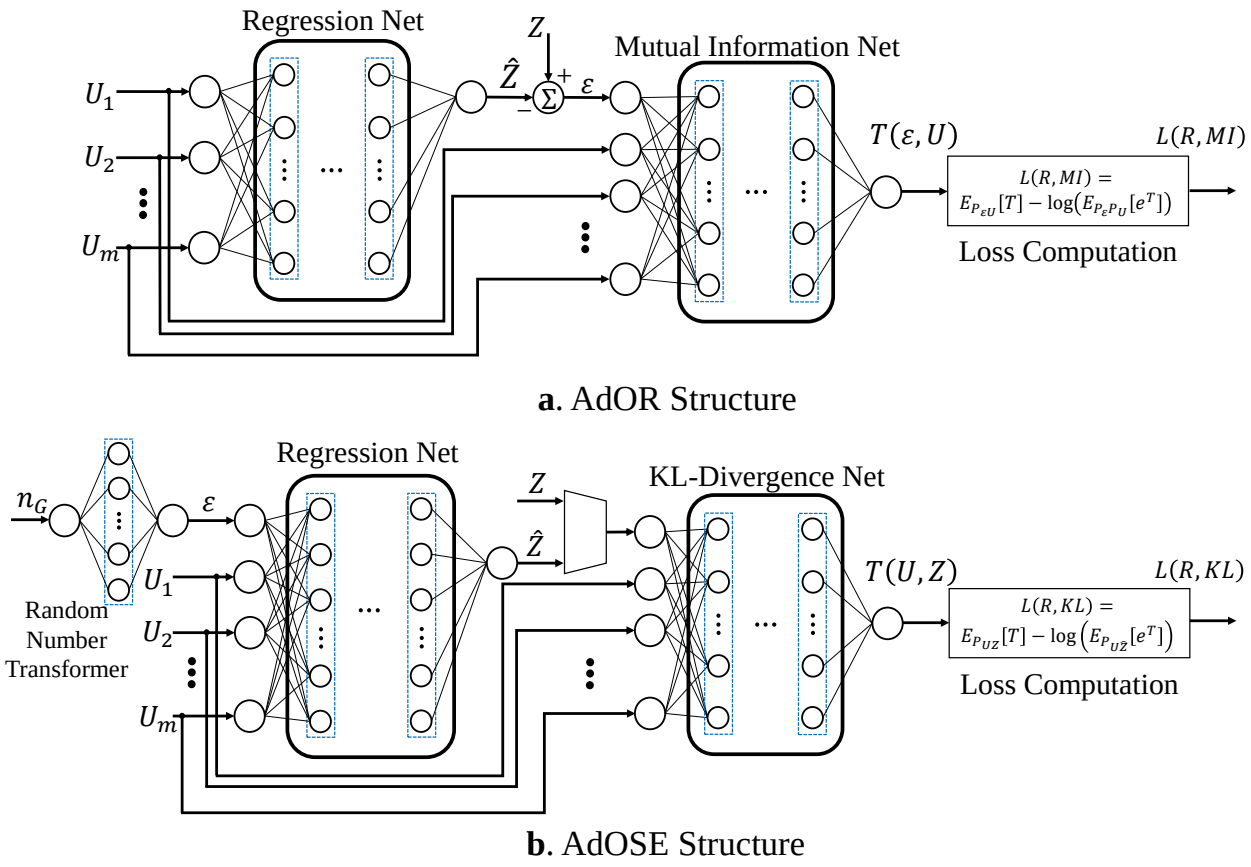


Figure 1: Block diagram of AdOR and AdOSE. **a**. AdOR structure: $(U_1, \ldots, U_m)$ are the input regressors, $\hat{Z}$ is the predict, and ε is the residual. The output loss $L(R, MI)$ is the estimated mutual information between them. **b**. AdOSE structure: n_G is generated by Gaussian generator, ε is the exogenous noise, $(U_1, \ldots, U_m)$ and ε are fed as inputs to R. KL computes the output twice: once using $(U_1, \ldots, U_m, Z)$, and once by $(U_1, \ldots, U_m, \hat{Z})$. The output $L(R, KL)$ is the KL-Divergence.

4. Adversarial Orthogonal Structural Equation Model

In (6), the noise ϵ is modeled as an additive term. However, in general, the exogenous noise can affect the variable Z in a non-linear form, such as in structural equation models (SEM, see [27]). Thus, we assume here that the true model is:

$$Z = f(U, \epsilon). \tag{9}$$

In AdOSE, we propose a new method to estimate both the nonlinear function f and also the joint distribution P_{UZ}. Hence, our goal is to obtain a function $\hat{f}$:

$$\hat{Z} = \hat{f}(U, \varepsilon), \tag{10}$$

such that $\hat{Z}$ is similar as possible as to the response Z, with the same U; i.e. $D_{KL}\left(P_{UZ} \middle|\middle| P_{U\hat{Z}}\right) = 0$.

In AdOSE, similar to AdOR, the regression network (R) is pitted against the loss network: a KL-divergence estimator (KL) that learns to match the joint distribution $P_{U\hat{Z}}$ to distribution P_{UZ} (see the bottom diagram of Figure 1). Inspired by GAN (Goodfellow et al. [7]), in AdOSE, the noise n_G is generated by a random Gaussian generator and transformed to the noise ε through a one-hidden layer perceptron $RanTrans$; i.e. $\varepsilon = RT\left(n_G\right)$. Then, regressors U and generated noise ε are passed to the regression network R, similar to AdOR; $\hat{Z} = \hat{f}\left(U, \varepsilon; \theta_{MI}\right)$. Afterwards, pairs (U, Z) and $(U, \hat{Z})$ are passed through KL by a differentiable transformation T, and the outputs are $T(U, Z; \theta_{MI})$ and $T(U, \hat{Z}; \theta_{MI})$, respectively. Based on (2), the KL-distance is estimated by $L(R, KL) = \mathbb{E}_{P_{UZ}}[T] - \log\left(\mathbb{E}_{P_{U\hat{Z}}}\left[e^T\right]\right)$, and two networks play the following minimax game:

$$\min_R \max_{KL} L(R, KL) = \mathbb{E}_{P_{UZ}}[T] - \log\left(\mathbb{E}_{P_{U\hat{Z}}}\left[e^T\right]\right). \tag{11}$$

At equilibrium, the value of loss $L(R, KL)$ is zero. After training, instead of having a nonlinear mapping between regressors and response, we have a nonlinear transformation for each samples of $U = u$, that assigns a distribution for

Z; i.e. $\hat{Z} \sim P(Z|U = u)$. Indeed, as the true value of n_G is unknown, we can not obtain single predict for each input sample u; while, we can draw output samples by feeding different values of n_G. Since training AdOSE is more trickier than AdOR, we provide some implementation details in Section 5.1 to avoid divergence of the algorithm.

Algorithm 2 shows the training procedure of AdOSE. In forward path, b Gaussian samples are drawn and fed to $RanTrans$. The regression output is computed in line 3. As MI in AdOR, KL evaluates T twice: once by using $u^{(i)}$ and true responses $z^{(i)}$, and once by $u^{(i)}$ and predicted responses $\hat{z}^{(i)}$ (line 4). Mini-batch loss L_B is then computed using mean of the true and estimated T. Similar to AdOR, in backward path, k_R and k_{KL} control the training of two networks. Furthermore, they play the main rule in convergence of the algorithm; if the loss is large, R has bad predicts and k_R should be increased, and if it is small, KL can not distinguish between true and predicted values and k_{KL} should be increased. Finally, note that $RanTrans$ is updated using gradients of its variables, concurrent with regression network R. So, $RanTrans$ can be viewed as a layer of R and one can obviously change the architecture of R without using $RanTrans$.

Algorithm 2: AdOSE

for *number of iterations* **do**

Forward path:

1. Generate b Gaussian samples $\{n_G^{(1)}, \dots, n_G^{(b)}\}$
   Feed them to $RanTrans$: $\varepsilon^{(i)} = RT(n_G^{(i)})$
2. Draw b minibatch examples
   $\left\{\left(u^{(1)}, z^{(1)}\right), \dots, \left(u^{(b)}, z^{(b)}\right)\right\}$
3. Evaluate regression output
   $\hat{z}^{(i)} = \hat{f}\left(\varepsilon^{(i)}, u^{(i)}; \theta_R\right); i = 1, \dots, b$
4. Evaluate output of KL twice
   $T^{(i)} = T\left(z^{(i)}, u^{(i)}; \theta_{KL}\right); i = 1, \dots, b$
   $T_{es}^{(i)} = T\left(\hat{z}^{(i)}, u^{(i)}; \theta_{KL}\right); i = 1, \dots, b$
5. Compute loss
   $L_B\left(\theta_R, \theta_{KL}\right) = \frac{1}{b}\sum_{i=1}^{b} T^{(i)} - \log\left(\frac{1}{b}\sum_{i=1}^{b} e^{T_{es}^{(i)}}\right)$

Backward path:

for k_R *steps* **do**
  Update R and $RanTrans$ by descending their stochastic gradients $\nabla_{\theta_R} L_B$ and $\nabla_{\theta_{RanTrans}} L_B$, respectively

end

for k_{KL} *steps* **do**
  Update KL by ascending its stochastic gradient $\nabla_{\theta_{KL}} L_B$

end

end

Applications in Causal Inference

AdOR and AdOSE can be used in causal models that assume there is a structural model between child and parents. For instance, consider the additive noise model (ANM) between the cause variable C and the effect variable E: $E = f(C) + \varepsilon$. In [12], it has been shown that there exist no function g and noise $\tilde{\varepsilon}$ almost surely such that $C = g(E) + \tilde{\varepsilon}$ and E and $\tilde{\varepsilon}$ are independent. Hence, we can utilize AdOR to infer causal direction between two variables X and Y. To do so, we regress each variable on the other one and pick the direction with minimum loss $L(R, MI)$. Moreover, one can use AdOR as the class of functions for TiMINO (Peters et al. [26]) for inferring causal direction in time series. At last, the causally conditioned DI [29] of each child on each candidate parent can also be estimated by regress the child twice, one on all variables, and the other on all variables except the candidate parent. The difference of two residuals' entropy is DI from parent to child.

5. Experiments

In this section, we first evaluate the performance of proposed regression methods on synthetic data and compare with the method in [22] and some other nonlinear regression methods. Then, we apply the proposed method to find the causal direction in some real-world bilinear data [23].

5.1. Implementation Details

The main point in training both AdOR and AdOSE is that the two networks R and MI (KL in AdOSE) should be trained simultaneously. As discussed before, Adam optimizer [17] is used, and all weights and biases initialized using Xavier initializer [6]. The number of layers, learning rate, and batch size are chosen similar in both networks.

In AdOR, we use three hidden layers with $tanh$, $sigmoid$ and $leaky{-}ReLU$ activation functions for R and three hidden layers with $leaky{-}ReLU$ activation for MI. Note that adding a bias term to $\hat{f}(U)$ in (7) does not change mutual information, so bias term is removed from output layer of R. Similarly, adding a constant term to $T\left(\varepsilon, U; \theta_{MI}\right)$ does not change the computed loss $L(R, MI)$ in (8), and we omit the bias term from output layer of MI. Instead, the maximum mini-batch value $\max_{i=1,\dots,b}\{T^{(i)}, T_{sh}^{(i)}\}$ is subtracted from whole $T^{(i)}$ and $T_{sh}^{(i)}$ in order to obtain a stable computation of loss.

The structure of AdOSE layers are designed similar to AdOR. The noise n_G is generated by normal Gaussian distribution, and $RanTrans$ has a hidden layer with $leaky{-}ReLU$ activation. The bias term is added to the output layer of R, and biases in KL are similar to MI. Finding the stable solution of AdOSE is more trickier than AdOR. The optimizer might diverge in the first few iterations, because one of networks R or KL outstrips the other. To avoid this, we adjust steps k_R and k_{KL} by looking at the value of loss L_B in each iteration in order to stabilize the training procedure. A simple choice of steps has a linear feedback form $k_R = \lfloor a + bL_B \rfloor$ and $k_{KL} = \lfloor a - bL_B \rfloor$ where a and b are some integer variables.

5.2. Toy Examples

In this part, AdOR and AdOSE are compared with eight regression methods: Support Vector Regression [33] (SVR),

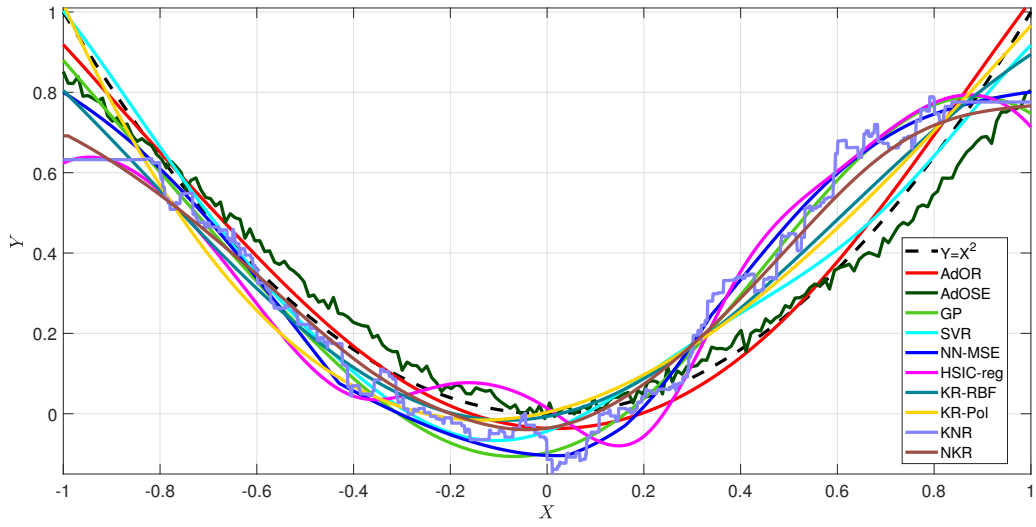


Figure 2: Example of different methods' outputs. $Y = X^2 + N$ and $N \sim F(8.5, 12.2)$. ($F(.,.)$ is F-distribution)

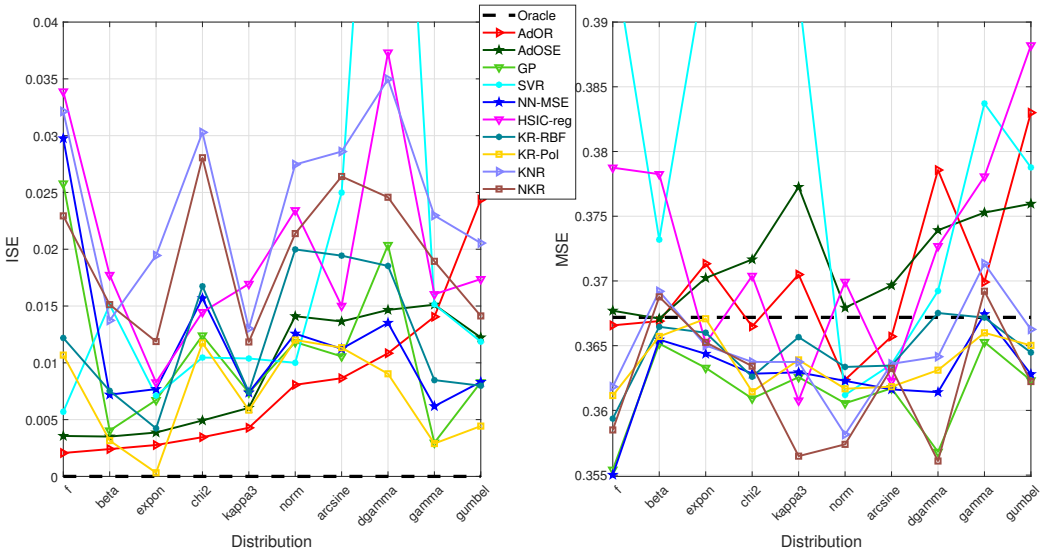


Figure 3: ISE (left) and MSE (right) of methods for the benchmark model $Y = X^2 + N$ for different distributions of noise N. (x-axis is sorted based on the ISE of AdOR.)

neural network with same structure as AdOR with MSE loss minimization (NN-MSE), HSIC regression proposed by Mooij et al. [22] (HSIC-reg), Gaussian Process regression [40] with RBF kernel (GP), Kernel Ridge regression [24] with RBF kernel (KR-RBF), again Kernel Ridge regression with polynomial degree 3 kernel (KR-Pol), k-Nearest Neighbor regression [1] with k=50 (KNR), and Nadaraya-Watson non-Parametric Kernel Regression [39, 25] with AIC Hurvich bandwidth estimation (NKR). The model has a simple form of $Y = f(X) + N = f(X) + kN'$. In each trial, 300 samples are drawn from uniform distribution $X \sim U(-1, 1)$. The function $f(.)$ is nonlinear and N' is generated from different distributions, and normalized by a constant k in order to have a similar signal-to-noise ratio $SNR = \mathbb{E}\left[f(X)^2\right] / \mathbb{E}\left[N^2\right] = 0.5$ for each trial. Figure 2 shows the output of different methods for the case of $f(X) = X^2$ and $N \sim F(8.5, 12.2)$. ($F(d_1, d_2)$ is F-distribution with parameters d_1 and d_2.) Note that for AdOSE, the averaged $\mathbb{E}_\epsilon[Y|X = x]$ is plotted by feeding 5000 samples of n_G at each $X = x$.

Comparison between methods for $f(X) = X^2$ is shown in Figure 3 for different distributions of N. More functions $f(X)$ with different noise distributions are compared in supplementary material. To evaluate methods, we measured both Mean Squared Error (MSE) between predictions and responses, and Integral Squared Error (ISE) between estimated function and $f(X)$. As can be seen, AdOR usually

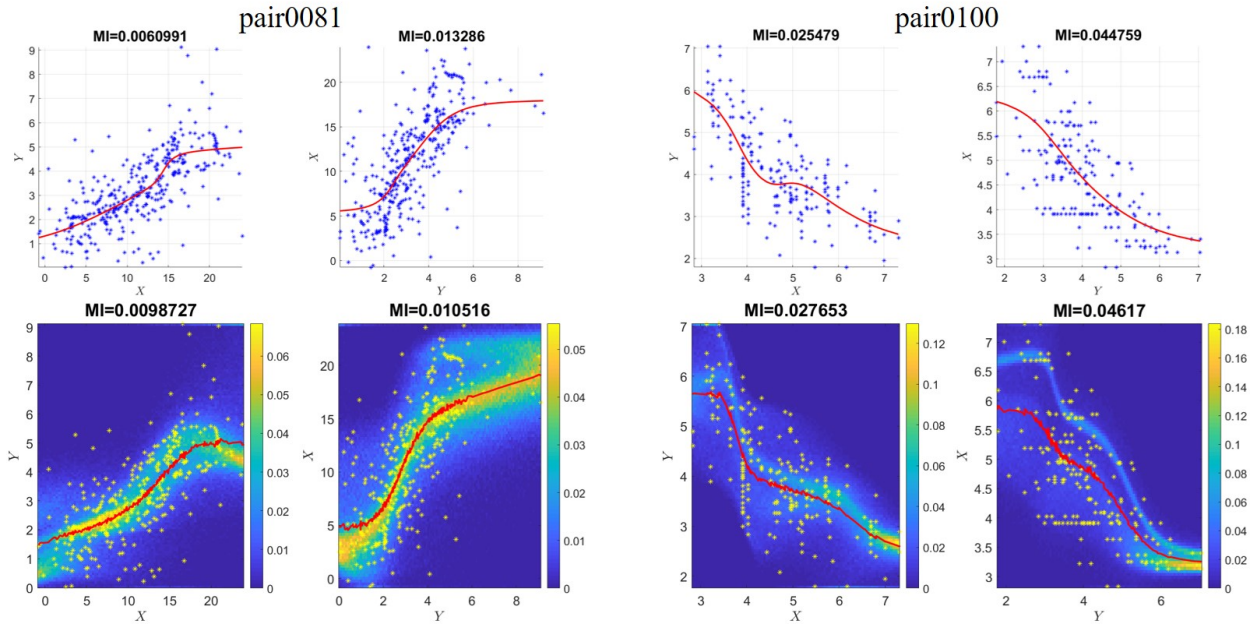


Figure 4: Results of proposed methods on real Tuebingen dataset. For both pairs, X is cause and Y is effect. **Top row**: AdOR **Bottom row**: AdOSE (heatmap shows posterior conditional probability $P(Y|X)$ at each $X = x$ and red trace is $\mathbb{E}_{\varepsilon}[Y|X = x]$.)

has the worst (highest) MSE among the others; in contrast, its performance is much better in terms of ISE measure. In fact, we expect that AdOR/AdOSE do not have better performance in terms of MSE, compared with regression methods minimizing squared losses (or similar losses) since the goal of such methods is actually to minimize MSE while AdOR tries to minimize the mutual information between the residual and regressors. Moreover, it is not guaranteed that regression methods with square loss error estimate the underlying function statistically efficiently in cases other than Gaussian additive noise. In such cases, although the squared loss is minimized, the result might be dependent on the regressors. In order to show that ISE is a better performance measure than MSE, suppose that there is a oracle that finds the truth function $f(X)$. Its ISE obviously is zero; while we see in right panel of Figure 3 that its MSE is larger than the some other methods. Therefore, lower MSE does not mean that we have a better approximation of f.

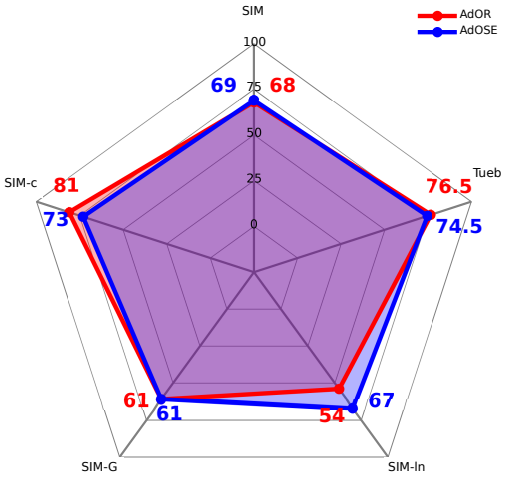


Figure 5: Rate of true discovered directions for each dataset

5.3. Causal Direction Discovery in Benchmark Datasets

In this part we examine AdOR and AdOSE in causal direction discovery task on five datasets based on the method of [12]. Following experiments part of [23], four scenarios are considered in order to create four synthetic datasets: *SIM* without confounders, *SIM-c* with a one-dimensional latent confounder, *SIM-ln* which has low level noise, and *SIM-G* with distributions close to Gaussian. Each dataset contains 100 pairs and there are 1000 samples in each pair, where we considered all of them. Additionally, `Cause-effect` pairs (Tuebingen dataset, version 1.0) [23] is a collection of 108 real-world pairs, each with different sample size from 94 to 16382, where we considered 102 number of these pairs. (The reminder 6 pairs are multivariate cases which removed from our evaluation.) Each pair of Tuebingen dataset consists of samples of two statistically dependent random variables X and Y, where one variable is known to causally influence the other. The task in this part is to infer which variable is the cause and which one is the effect.

AdOR is trained with each pair of each dataset twice: once when Y is response and X is regressor and once in the reverse direction. The direction with lower mutual information is considered as the true direction. In the same manner, AdOSE is trained twice in forward and reverse directions.

Table 1
Accuracy of AdOR, AdOSE, and baseline methods. The last row includes AUC of Tuebingen benchmark.

DATA SET	ADOR	ADOSE	ANM	IGCI	RESIT	RECI	CURE	PNL	GPI	QCCD
SIM (ACC%)	68	69	76	38	72	70	57	70	82	61
SIM-c (ACC%)	81	73	85	47	82	74	63	65	86	72
SIM-ln (ACC%)	54	67	74	62	87	80	62	61	88	73
SIM-G (ACC%)	61	61	76	85	72	65	50	64	94	62
TUEB (ACC%)	76.5	74.5	70	63	72	70	73	75	67	67
TUEB (AUC)	0.71	0.68	0.73	0.72	0.70	0.70	0.66	0.70	0.61	0.71

The estimated function $\hat{f}(x) = \mathbb{E}_{\epsilon}[Y|X = x]$ is computed by feeding 5000 samples of n_G at each $X = x$. Figure 4 shows the estimated functions of AdOR and AdOSE on two pairs `pair0081` and `pair0100` of `Cause-effect` pairs. The results for other pairs are given in the supplementary material.

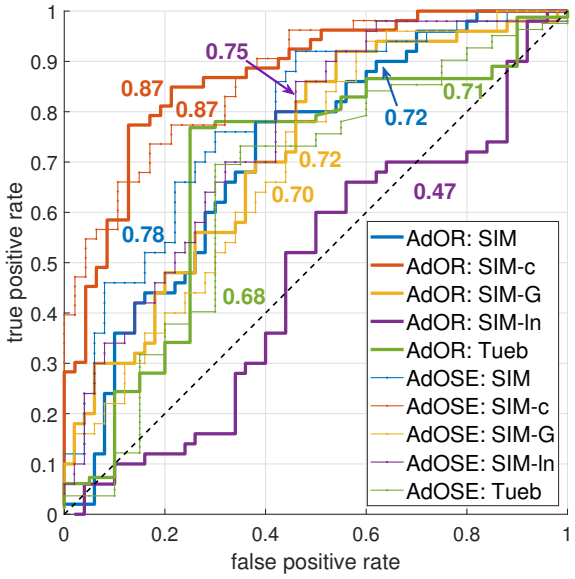


Figure 6: ROC curves of AdOR and AdOSE for each dataset. AUC for each dataset is written close to the corresponding curve.

We evaluated our methods based on two metrics: the rate of true discovered directions (Figure 5), and AUC (Area Under ROC Curve, Figure 6). In order to obtain ROC curves, we defined the score $S_i = MI(\varepsilon_{Y\to X}, Y) - MI(\varepsilon_{X\to Y}, X)$ for each pair i. The inferred direction is true for pair i if $S_i > 0$. Also, as proposed in [23], we plotted correct decision rate versus decision rate for `Cause-effect` pairs. The result is shown in Figure 7. It shows how much we can trust on the scores S_i; i.e., whether a score with higher absolute value is more reliable than the lower one. Mathematically, we consider a positive threshold th. Then, we define: $DecisionRate = \frac{card(\{S_i|abs(S_i)\geq th\})}{d}$, $CorrectDecision = \frac{card(\{S_i|abs(S_i)\geq th, S_i\geq 0\})}{card(\{S_i|abs(S_i)\geq th\})}$. ($card(.)$ denotes the cardinality of a set, $abs(.)$ is absolute function, and $d = 102$ is the size of dataset.) By Changing th values, different points of Figure 7 are obtained.

Before applying our methods on each pair, we applied a pre-processing step on data: out of range samples, samples that are lower than 5% or higher than 95% percentile, were removed from each pair for both X and Y samples. Experiments were done with and without this pre-processing step. We found that this step improve the results: on Tuebingen dataset, AdOR score increased from $ACC = 70\%, AUC = 0.67$ to $ACC = 76.5\%, AUC = 0.71$, and AdOSE scores increased from $ACC = 67\%, AUC = 0.62$ to $ACC = 74.5\%, AUC = 0.68$. Also, removing out of range samples make AdOSE more stable in training phase. Intuitively, out of range samples have a huge cost because of exponential term in estimator 4. Hence, we recommend apply this pre-processing step when testing the code on the other datasets.

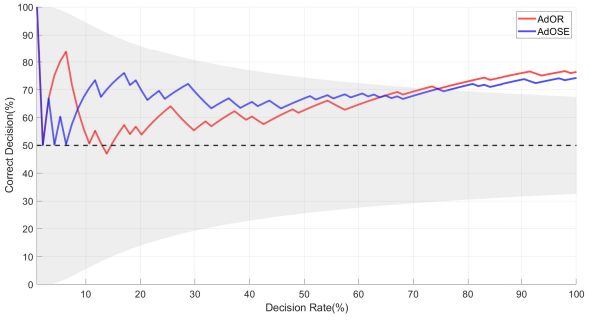


Figure 7: Correct decisions rate versus decision rate for AdOR and AdOSE on Tuebingen benchmark. The gray area shows the 95% confidence interval of a random coin flip.

The results of the other approaches in the following list are compared with our method in Table 1: ANM [23], IGCI [14], RESIT [28], RECI [3], CURE [30], PNL [43], GPI [34], and QCCD [35]. As can be seen, AdOR and AdOSE are among the top algorithms in almost all datasets; remark-

ably, our methods outperform competitors in benchmark Tuebingen dataset.

6. Conclusions

We introduced two novel regression methods: AdOR which minimizes mutual information between the residual and the regressors, and AdOSE which produce response that mimics the true output by reducing distance between joint distributions. Conducted with details, we implemented our methods through adversarial neural networks and showed their great potential for non-parametric regression and further for inferring causal influences in models with unknown noise distributions. As a future work, one can extend these methods to the cases with categorical variables or utilize them in other causal learning problems such as learning causal structures.

References

[1] Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46, 175–185.

[2] Belghazi, M.I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, R.D., 2018. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062 .

[3] Bloebaum, P., Janzing, D., Washio, T., Shimizu, S., Schölkopf, B., 2018. Cause-effect inference by comparing regression errors, in: International Conference on Artificial Intelligence and Statistics, pp. 900–909.

[4] Darbellay, G.A., Tichavsky, P., 2000. Independent component analysis through direct estimation of the mutual information, in: ICA, pp. 69–75.

[5] Donsker, M.D., Varadhan, S.S., 1983. Asymptotic evaluation of certain markov process expectations for large time. iv. Communications on Pure and Applied Mathematics 36, 183–212.

[6] Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256.

[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

[8] Granger, C.W., 1969. Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society , 424–438.

[9] Granger, C.W.J., 1963. Economic processes involving feedback. Information and control 6, 28–48.

[10] Haury, A.C., Mordelet, F., Vera-Licona, P., Vert, J.P., 2012. Tigress: trustful inference of gene regulation using stability selection. BMC systems biology 6, 145.

[11] Hausser, J., Strimmer, K., 2009. Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. Journal of Machine Learning Research 10, 1469–1484.

[12] Hoyer, P.O., Janzing, D., Mooij, J.M., Peters, J., Schölkopf, B., 2009. Nonlinear causal discovery with additive noise models, in: Advances in neural information processing systems, pp. 689–696.

[13] James, W., Stein, C., 1992. Estimation with quadratic loss, in: Breakthroughs in statistics. Springer, pp. 443–460.

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

[15] Jazayeri, M., Afraz, A., 2017. Navigating the neural space in search of the neural code. Neuron 93, 1003–1014.

[16] Kay, S.M., 1993. Fundamentals of statistical signal processing. Prentice Hall PTR.

[17] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[18] Liu, Y., Aviyente, S., Al-khassaweneh, M., 2009. A high dimensional directed information estimation using data-dependent partitioning, in: 2009 IEEE/SP 15th Workshop on Statistical Signal Processing, IEEE. pp. 606–609.

[19] Marbach, D., Costello, J.C., Küffner, R., Vega, N.M., Prill, R.J., Camacho, D.M., Allison, K.R., Aderhold, A., Bonneau, R., Chen, Y., et al., 2012. Wisdom of crowds for robust gene network inference. Nature methods 9, 796.

[20] Marko, H., 1973. The bidirectional communication theory-a generalization of information theory. IEEE Transactions on communications 21, 1345–1351.

[21] Miller, E.G., 2003. A new class of entropy estimators for multi-dimensional densities, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., IEEE. pp. III–297.

[22] Mooij, J., Janzing, D., Peters, J., Schölkopf, B., 2009. Regression by dependence minimization and its application to causal inference in additive noise models, in: Proceedings of the 26th annual international conference on machine learning, ACM. pp. 745–752.

[23] Mooij, J.M., Peters, J., Janzing, D., Zscheischler, J., Schölkopf, B., 2016. Distinguishing cause from effect using observational data: methods and benchmarks. The Journal of Machine Learning Research 17, 1103–1204.

[24] Murphy, K.P., 2012. Machine learning: a probabilistic perspective. MIT press.

[25] Nadaraya, E.A., 1964. On estimating regression. Theory of Probability & Its Applications 9, 141–142.

[26] Peters, J., Janzing, D., Schölkopf, B., 2013. Causal inference on time series using restricted structural equation models, in: Advances in Neural Information Processing Systems, pp. 154–162.

[27] Peters, J., Janzing, D., Schölkopf, B., 2017. Elements of causal inference: foundations and learning algorithms. MIT press.

[28] Peters, J., Mooij, J.M., Janzing, D., Schölkopf, B., 2014. Causal discovery with continuous additive noise models. The Journal of Machine Learning Research 15, 2009–2053.

[29] Quinn, C.J., Kiyavash, N., Coleman, T.P., 2015. Directed information graphs. IEEE Transactions on information theory 61, 6887–6909.

[30] Sgouritsa, E., Janzing, D., Hennig, P., Schölkopf, B., 2015. Inference of cause and effect with unsupervised inverse regression, in: Artificial intelligence and statistics, pp. 847–855.

[31] Shadlen, M.N., Britten, K.H., Newsome, W.T., Movshon, J.A., 1996. A computational analysis of the relationship between neuronal and behavioral responses to visual motion. Journal of Neuroscience 16, 1486–1510.

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

[33] Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14, 199–222.

[34] Stegle, O., Janzing, D., Zhang, K., Mooij, J.M., Schölkopf, B., 2010. Probabilistic latent variable models for distinguishing between cause and effect, in: Advances in neural information processing systems, pp. 1687–1695.

[35] Tagasovska, N., Vatter, T., Chavez-Demoulin, V., 2018. Nonparametric quantile-based causal discovery. arXiv preprint arXiv:1801.10579 .

[36] Ver Steeg, G., Galstyan, A., 2012. Information transfer in social media, in: Proceedings of the 21st international conference on World Wide Web, ACM. pp. 509–518.

[37] Ver Steeg, G., Galstyan, A., 2013. Information-theoretic measures of influence based on content dynamics, in: Proceedings of the sixth ACM international conference on Web search and data mining, ACM. pp. 3–12.

[38] Wang, Z., Bovik, A.C., 2009. Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE signal processing magazine 26, 98–117.

[39] Watson, G.S., 1964. Smooth regression analysis. Sankhyā: The Indian Journal of Statistics, Series A , 359–372.
[40] Williams, C.K., Rasmussen, C.E., 1996. Gaussian processes for regression, in: Advances in neural information processing systems, pp. 514–520.
[41] Zellner, A., 1988. Causality and causal laws in economics. Journal of econometrics 39, 7–21.
[42] Zhang, K., Hyvärinen, A., 2009. Causality discovery with additive disturbances: An information-theoretical perspective, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 570–585.
[43] Zhang, K., Hyvarinen, A., 2012. On the identifiability of the post-nonlinear causal model. arXiv preprint arXiv:1205.2599 .