Post-nonlinear Causal Model with Deep Neural Networks

Youngseog Chung (youngsec) [1] **Joon Kim (joonsikk)** [1] **Tom Yan (tyan2)** [1] **Helen Zhou (hlzhou)** [1]

Abstract

Causal discovery is a process of discovering causal information from observed data. Previous work in the field has focused on discovering causal directions in two variables, with extra effort on constructing assumptions for the data distribution and proving identifiability. However they lack flexibility in scaling up causal discovery for multiple variables and a systematic end-to-end pipeline. We extend two existing work in the literature (LiNGAM and PNL), and by integrating them with deep neural networks, we propose an end-to-end learning scheme for training a model for causal discovery. We evaluate our method on several real and simulated datasets.

1. Introduction

Data analysis is often driven by causal questions. A biostatistician may be interested in the effectiveness of a treatment, a financial analyst may be interested in the effect of government actions on a stock market crash, and an insurer may want to know what costs can be attributed to obesity. However, these types of questions are not easily expressed in the traditional language of statistics (Pearl et al., 2016). This motivates our work, which offers a possible approach for causal discovery.

Causal discovery refers to the discovery of causal information from observational data. This is different from but closely related to *causal inference*, which involves finding the causal effect of one variable on another. Previous causal discovery algorithms have exploited conditional independence tests for removing unnecessary connections among the observed variables, in order to produce a set of acyclic causal models which are in the d-separation equivalence class.

This work investigates and extends the post-nonlinear (PNL) acyclic functional causal model (Zhang & Hyvärinen, 2009). In this model, each observed variable is assumed to be generated by a nonlinear function of its parents, with additive noise, followed by a nonlinear distortion. The nonlinearity in the second stage takes into account the effect of sensor distortions, which are usually encountered in practice. With this assumption regarding the form of the functional causal model, it is possible to infer causal relationships between any pairs of variables.

As we will show, this can be extended to jointly infer the causal relationships between all variables in an end-to-end fashion. We find this direction appropriate as scalability is a major challenge in causal discovery and an end-to-end solution allows for more efficient structural discovery. This can lead to some complicated structures, however, so we also explore a simplified model which may be more amenable to practical use. Finally, we evaluate our models on both real and simulated datasets.

To summarize, the contributions of this work are: (1) jointly inferring the causal directions among multiple variables in a scalable end-to-end fashion, (2) simplifying the model for easier practical use and interpretation, and (3) evaluating our models on both real and simulated datasets.

2. Background

Following Zhang & Hyvärinen (2010), this work uses post-nonlinear functions to model causal relationships. PNL functions are defined to have the following form:

$$x_i = g_i(f_i(pa_i) + e_i)$$

where f_i denotes the nonlinear effect of the causes, g_i denotes an invertible post-nonlinear distortion, pa_i is the parent cause of x_i, and e_i is an independent disturbance. Note that pa_i is independent of e_i, but x_i is not. Rearranging, the noise terms can be recovered by using the inverse of g_i:

$$e_i = g_i^{-1}(x_i) - f_i(pa_i)$$

Now, suppose we are curious about the causal relationship between two variables x_1 and x_2. If $x_1 \to x_2$ (i.e. $x_i = x_2$ and $pa_i = x_1$ in the above equations), then as previously mentioned, the parent cause x_1 should be independent of the noise e_2. That is, there exist nonlinear functions g_1^{-1}, g_2^{-1}, and f_2 such that $g_1^{-1}(x_1)$ is independent of $e_2 = g_2^{-1}(x_2) - f_2(x_1)$.

[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

Using this key observation, our goal is now to see whether there exist such nonlinear functions that make $g_1^{-1}(x_1)$ and e_2 independent. In our work, neural networks are used to parameterize these nonlinear functions, and we optimize for independence by minimizing mutual information between x_1 and e_2. Previous work (Almeida, 2003) has utilized constrained nonlinear independent component analysis (ICA) followed by statistical independence tests to distinguish the cause from the effect in the two-variable case.

Often, however, we are interested in causal interactions between not just two variables, but *many* variables. One could try to exhaustively test all possible causal relationships between pairs of variables, but the computational costs of this grows exponentially. Thus, the direction we explore builds off of ideas described by (Zhang & Hyvärinen, 2009) in order to extend this framework to the multivariate case. The goal is to jointly infer causal relationships between many variables in a scalable manner, leveraging insights from the two-variable case.

Luckily, previous work has shown that the PNL model is *identifiable* in all but five cases which are listed in Table 1 of (Zhang & Hyvärinen, 2009). We provide the definition and theorem for identifiability below.

Definition 4 (Identifiability) (Pearl, 2009) The causal effect of X on Y is said to be identifiable if the quantity $P(y|x)$ can be computed uniquely from any positive distribution of the observed variables, that is, if for every pair of theories T_1 and T_2 such that $P_{T1}(v) = P_{T2}(v) > 0$, we have $P_{T1}(y|x) = P_{T_2}(y|x)$.

That is, under identifiability, for any causal models which assign equal and positive probability to the set of observed variables, the inferred causal effects will be the same. This is desirable because models which maximize likelihood should not give conflicting inferences.

Now, Theorem 1 (Zhang & Hyvärinen, 2009) provided below shows that mutual independence of the noise terms e_i is a sufficient condition for identifiability.

Theorem 1. *When fitting variables x_1, ..., x_n to the PNL acyclic causal model with the causal structure represented by the DAG G, the noise terms e_i are mutually independent if and only if the causal Markov condition holds (i.e. each variable x_i is independent of its non-descendants conditional on its parents in G), and the noise e_i in x_i is independent of the parents of x_i.*

Proof. A sketch goes as follows: We only show $\Rightarrow$ direction as the other direction follows by definition of PNL models.

We want to show that $e_1, .., e_n$ are mutually independent, or equivalently, $x_1, ..., x_n$ follow the PNL causal model represented by G, if the causal Markov condition holds and the noise e_i is independent of Pa_i.

Write $z_i = g_i^{-1}(x_i)$. As the causal relations are acyclic, we can obtain a topological order such that no later variable causes any earlier one. Let this order be $(x_1, ..., x_n)$.

By change of variables from $z_i \to x_i$ with $x_i = g_i(z_i)$:

$p(x_i|\text{Pa}_i) = p(z_i|\text{Pa}_i)/|g_i'(z_i)|$

Therefore:

$$\begin{aligned} H(e_i) &\geq H(e_i|\text{Pa}_i) \\ &= H(z_i|\text{Pa}_i) \\ &= -\mathbb{E}[\log p(z_i|\text{Pa}_i))] \\ &= -\mathbb{E}[\log p(x_i|\text{Pa}_i))] - \mathbb{E}[\log |g_i'(z_i)|] \\ &= H(x_i|\text{Pa}_i) - \mathbb{E}[\log |g_i'(z_i)|] \\ &\geq H(x_i|x_1, ..., x_{i-1}) - \mathbb{E}[\log |g_i'(z_i)|] \end{aligned}$$

Summing this across i's to get:

$$\begin{aligned} \sum_i H(e_i) &\geq \sum_i H(x_i|x_1, ..., x_{i-1}) - \sum_i \mathbb{E}[\log |g_i'(z_i)|] \\ &= H(x_1, ..., x_n) - \sum_i \mathbb{E}[\log |g_i'(z_i)|] \end{aligned}$$

Equality on holds if e_i is independent of Pa_i and that the causal markov condition holds, meaning $\{x_k|x_k \notin \text{Pa}_i, k \in [i-1]\}$ are independent of x_i given Pa_i.

If the causal markov condition holds and e_i is independent of Pa_i , consider the transformation from $(x_1, .., x_n) \to (e_1, .., e_n)$, the Jacobian J is lower-triangular since e_i does not depend on x_j for $(j > i)$. $J_{i,i} = 1/g_i'(z_i)$. Therefore, we have the determinant $|J| = (\prod_i g_i'(z_i))^{-1}$.

Therefore, if the causal markov condition holds, $\sum_i H(e_i) = H(x_1, ..., x_n) - \sum_i \mathbb{E}[\log |g_i'(z_i)|]$ and the mutual information of $e_1, ..., e_n$ is such that the below equality holds:

$$\begin{aligned} I(e_1, .., e_n) &= \sum_i H(e_i) - H(e_1, ..., e_n) \\ &= \sum_i H(x_i) - [H(x_1, ..., x_i) + \mathbb{E}[\log |J|]] \\ &= \sum_i H(e_i) - [H(x_1, ..., x_i) + \sum_i \mathbb{E}[\log |g_i'(z_i)|]] \\ &= 0 \end{aligned}$$

This implies $e_1, .., e_n$ are mutually independent.

□

3. Related Work

Zhang & Hyvärinen (2010) is most closely related to our proposed work. In this work, the authors model f_i's using multi-layered perceptrons (MLP) and exploit a two-step process to discover causal relationship between two variables. The first step is performing gradient descent to learn the parameters of the MLPs by minimizing the mutual information between e_i and pa_i. The second step is performing a statistical independence test to see if they are independent.

In (Zhang & Hyvärinen, 2009), the authors refine their work in Zhang & Hyvärinen (2010) by further investigating the identifiability of the PNL causal model. Notably, they enumerate all possible situations in which the PNL system is not identifiable, and also describe a possible way to extend the method of discovering causal relationships between two variables to multiple variables.

Several works justify minimizing dependency between the regressor and the noise for causal discovery. Mooij et al. (2009) suggest a framework for causal discovery that minimizes the dependency of the noise and the regressor, and use the Hilbert-Schmidt Independence Criterion (HSIC) test for verifying the independence, which resulted in a good performance for the NIPS Causality Challenge. Zhang et al. (2016) further justify the use of mutual information between the noise variables as the loss for the optimization in search for the parameters of the PNL causal model. It also introduces a new family of nonparametric methods for estimating the PNL causal model.

Combining causal models with deep learning has been an active area of research. Louizos et al. (2017a) use variational autoenconders to tackle the problem of discovering latent representation of the confounders from observed noisy version of the confounders. The authors make use of deep neural networks to encode/decode the distribution of these confounders, with a new architecture suited for addressing counterfactual queries. Unlike this work which focuses on learning representations of the true confounders using the observed proxies, we are more interested in discovering the true causal model using the observed variables based on the PNL model.

In addition to combining causal models, recent work has studied how to combine multiple datasets in order to improve learning in causal models. For many real-world problems, there are many data sources encoding the same underlying phenomena. Leveraging the presence of nonstationary heterogeneous data, Zhang et. al. proposed constraint-based causal discovery procedures (Zhang et al., 2017). While non-stationary data often changes the correlations that appear within data, this work observes that nonstationarity actually helps determine causal orientations due to the invariance of causal mechanisms.

In terms of application, prior works (Tran et al., 2017) have sought to study causal relationships in Genome-Wide Association Studies (GWAS). The work has sought to address two key issues in modeling GWAS. The first is latent confounders due to population structure. To address this, this paper suggests learning the confounders jointly with the rest of the model. The second problem is the highly nonlinear interactions between different parts of the genome. To address this, an "implicit causal model" is proposed and utilizes deep neural networks with implicit density to capture this rich nonlinearity. Inference is performed using likelihood-free variational inference.

Previous applied work has also aimed to construct causal biological networks by utilizing the active learning paradigm. Cho et al. (2016) learn Gaussian Bayesian networks from observational and intervention data, iteratively acquiring new data instances by carrying out the optimal interventions predicted to cause the largest change in belief, and updating the network accordingly. Their methods lead to significant runtime improvements, and are effective on both simulated data and the DREAM4 network inference challenge data sets.

4. Methods

We present two architectures for an end-to-end causal discovery: (1) Deep PNL, and (2) LinPNL which is a linear simplification of Deep PNL.

4.1. Deep PNL

To alleviate the issues of scalability and end-to-end learning, we propose Deep PNL, which parameterizes the nonlinearities as multi-layer perceptrons (MLPs) and trains the whole model end-to-end with an appropriate loss. We follow the standard post-nonlinear assumption that

$$e_i = g_i^{-1}(x_i) - f_i(pa_i), \qquad i \in \{1, ..., d\}$$

where the nonlinearities $g_i^{-1}(\cdot)$ and $f_i(\cdot)$ are modeled as MLPs. If we consider the two-variable case where x_2 causes x_1, then (as explained in Section 2) in the subnetwork shown in Figure 1, e_1 and e_2 should be independent for some f_i's and g_i's.

Now, we extend this to the d-variable case. Figure 2 shows the full model architecture given d-dimensional inputs. We treat pa_i as *all* variables other than x_i (as shown from the almost-fully-connected first layer), because prior to learning we do not know which variable causes which. We also impose L1 regularization on the objective function to encourage sparsity of the weights, pushing non-meaningful connections towards zero.

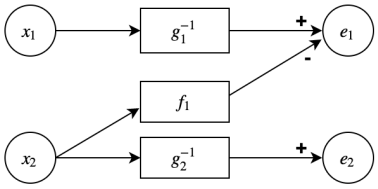

Figure 1. Sub-network of Deep PNL for when x_2 causes x_1.

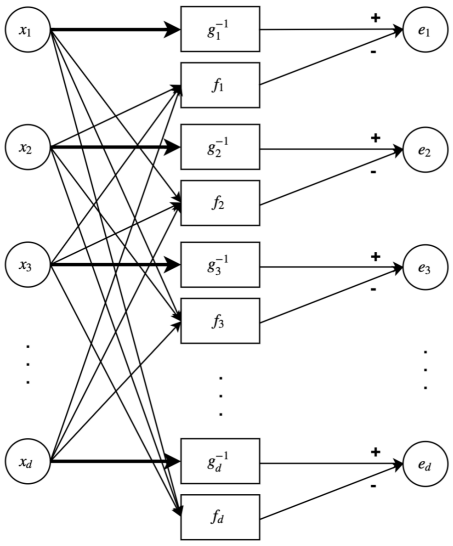

Figure 2. Deep PNL architecture for d variables. We model functions g_i^{-1}'s and f_i's with MLPs. The objective is to make the noise output e_i's independent, which is done by minimizing the mutual information $I(e_1, ..., e_d)$.

With the Deep PNL architecture, we minimize the mutual information of the noise terms, $I(e_1, ..., e_d)$, to make them as independent as possible. Formally, with Deep PNL model denoted as $\hat{\mathbf{e}} = \Phi_\theta(\mathbf{x})$, our objective is

$$\min_\theta I(\Phi_\theta(\mathbf{x})) + \lambda\|\theta\|_1.$$

Numerically optimizing the mutual information is not trivial, and we employ the tricks used in several previous works. One observation Zhang & Hyvärinen (2009) makes is that we can express the mutual information as

$$\begin{aligned} &I(e_1,...,e_d) \\ &\quad= \sum_{i=1}^{d} H(e_i) - H(e_1, ..., e_d) \\ &\quad= \sum_{i=1}^{d} H(e_i) - (H(x_1, ..., x_d) + \mathbb{E}[\log|\det J|]) \end{aligned}$$

where J is the Jacobian of e_i, e.g. $\left[\frac{\partial e_i}{\partial x_j}\right]_{ij}$. We can drop the second term from the objective because it does not depend on the parameter we are optimizing. But even with this, it is not clear how to compute the individual entropy terms as we do not assume anything about the distribution of e_i's. To do so, one can think about assuming some form of distribution of e_i's using Gaussian mixtures and learn the parameters jointly with the Deep PNL. Another method is to use the intuitions employed in INFOMAX (Linsker, 1988) or MISEP (Almeida, 2003). To minimize the mutual information among e_i's, we introduce nonlinear transformations, ψ_i's, for each variable, which is designed to model the CDF of e_i's, as shown in Figure 3.

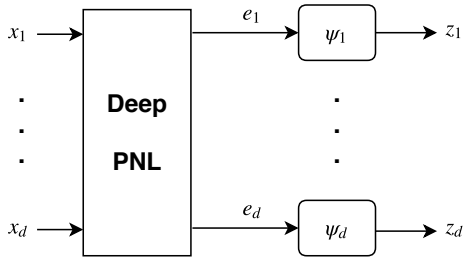

Figure 3. Architecture for training Deep PNL using INFOMAX/MISEP. We train ψ_i to be the CDF of e_i by maximizing the joint entropy $H(z_1, ..., z_d)$.

We make an observation that if ψ_i's are indeed CDFs of e_i's, then $z_i = \psi_i(e_i)$ will follow a uniform distribution $\mathcal{U}(0, 1)$. Then the individual entropy terms of z_i's will be zero, $H(z_i) = 0$, meaning that we have

$$\begin{aligned} &I(e_1, ..., e_d) = I(z_1, ..., z_d) \\ &\quad= \sum_i H(z_i) - H(z_1, ..., z_d) = -H(z_1, ..., z_d). \quad (1) \end{aligned}$$

Therefore our objective of minimizing $I(e_1, ..., e_d)$ is equivalent to maximizing $H(z_1, ..., z_d)$. Also note that maximizing $H(z_1, ..., z_d)$ with e_i's being fixed is equivalent to maximizing the individual entropy terms $H(z_i)$ as seen in Equation 1, leading to $z_i \sim \mathcal{U}(0, 1)$ as uniform distribution is the distribution that maximizes the entropy in a bounded support. As a result ψ_i we learn will indeed approach the CDF of e_i.

As a result, our revised objective for minimizing $I(e_1, ..., e_d)$ will be

$$\min_\phi -H(\mathbf{z}) = \min_\phi -\mathbb{E}[\log|\det J|]$$

where $J = \left[\frac{\partial z_i}{\partial x_j}\right]_{ij}$, and now we optimize over all the parameters ϕ which includes θ for the Deep PNL architecture as well as parameters for each ψ_i. For the implementation, we modeled ψ_i using MLP with sigmoid nonlinearities and non-negative weights to ensure that it is an increasing function.

4.2. A Linear Simplification to the Post-Nonlinear Functional Causal Model

4.2.1. LinPNL

One drawback of the Deep PNL model is that there are many parameters to be learned from a relatively small dataset. Additionally, it is usually hard to find large-scale causal datasets with known ground truth causal relationships. Under such settings, Deep PNL may perform suboptimally. To mitigate these issues and aim for simplicity, we propose a linear simplification to the post-nonlinear model for causal discovery, which we will refer to as LinPNL.

The key simplification of LinPNL is that the underlying causal relationship is linear. LinPNL still allows a nonlinear distortion to the raw variables, which may cause the observed data to be non-linear, but it is assumed that once these non-linear distortions are reversed, the relationship between cause and effect is linear.

As an illustration, consider the 2 variables case with x_1 and x_2, and suppose x_1 causes x_2. LinPNL assumes that x_1 and x_2 may not be linear due to some invertible, nonlinear distortion (e.g. due to observation error), modelled by g_1 and g_2. However, once we reverse these distortions, $g_1^{-1}(x_1)$ and $g_2^{-1}(x_2)$ are linear, i.e. $g_2^{-1}(x_2) = b_1 g_1^{-1}(x_1) + e_1$, where b_1 is a constant and e_1 is assumed to be non-Gaussian noise.

When we expand this model to d variables, the LinPNL model assumes that the data is generated from the expression $\mathbf{g}^{-1}(\mathbf{x}) = \mathbf{B}\mathbf{g}^{-1}(\mathbf{x}) + \mathbf{e}$, where $\mathbf{g}^{-1}(\mathbf{x})$ signifies element-wise application of the respective inverse functions, g_i^{-1} on x_i, $i \in \{1, ..., d\}$, and the matrix $\mathbf{B}$ encodes the linear causal relationships. Our objective in LinPNL is then to discover $\mathbf{B}$ from the data.

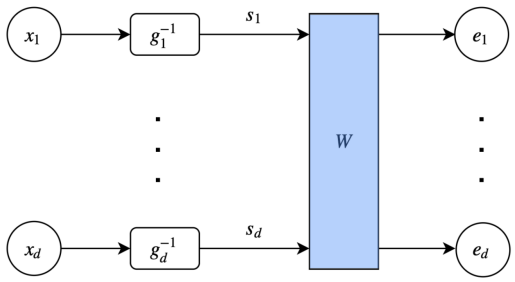

Figure 4. Linear simplification to the post-nonlinear causal model (LinPNL). After a nonlinear transformation, linear causal relationships are encoded in the matrix $W = I - B$.

To do so, we consider an architecture as shown in Figure 4. g_i's are nonlinear functions modelling some nonlinear distortion, and $\mathbf{W}$ is a linear mapping from s to e, i.e. $\mathbf{e} = \mathbf{W}\mathbf{s}$. Based on our assumption about the data generation, we have $\mathbf{e} = (\mathbf{I} - \mathbf{B})\mathbf{s}$, therefore learning this $\mathbf{W}$ will allow the recovery of $\mathbf{B}$. To learn $\mathbf{W}$, again, we want to minimize the mutual information of e_i's, and this can, again, be done with the methods used for Deep PNL, i.e. using the CDF approximators ψ_i's. In this case we would also like to add additional constraints on $\mathbf{W}$, e.g. sparsity, and diagonal elements being 1, to effectively learn the causal relationship from given data.

4.2.2. Comparison of LinPNL to an Existing Method

Some previous work has been done in discovering causal directions under the assumption of linear causal relationships. One of the most notable works is LiNGAM (Shimizu et al., 2006).

The assumptions of LiNGAM are almost identical to the assumptions of LinPNL, except LiNGAM does not model the nonlinear distortion, g. Hence, LiNGAM assumes the observed variables are linear in their causal relationships, $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}$, with $\mathbf{e}$ being non-Gaussian noise.

LiNGAM recovers the causal relationships by directly applying ICA on the observed data, $\mathbf{x}$, to derive the linear transformation matrix $\mathbf{W}$, which transform the data, $\mathbf{x}$ into a matrix with independent components as rows. The linear causal matrix $\mathbf{B}$ is recovered by an exhaustive row permutation on $\mathbf{I} - \mathbf{W}$ to make it as lower triangular as possible, which is effectively reordering the variables in $\mathbf{x}$ in a causal order so that effect variables always come after the cause variables in $\mathbf{x}$.

Comparing LinPNL to LiNGAM, LinPNL is more flexible than LiNGAM by allowing a nonlinear distortion on the data. Also, the computation load of LiNGAM grows exponentially with the number of variables because it performs an exhaustive permutation search on the causal relationship matrix. In comparison, LinPNL would scale much better with more variables because the number of parameters in the model network would only increase linearly with the number of variables.

Table 1 summarizes the models discussed so far and their corresponding assumptions.

Model	Functional Causal Assumption when $x_i \to x_j$ (note: B is a scalar)
LiNGAM	$x_j = B \cdot x_i + e_i$
LinPNL	$g_j^{-1}(x_j) = B \cdot g_i^{-1}(x_i) + e_i$
Deep PNL	$x_j = g_i(f_i(x_i) + e_i)$

Table 1. Summary of models and their assumptions about the functional form of causal relationships.

5. Experiments

Empirical analysis is done on real and simulated datasets. In this section, we provide practical implementation details, followed by evaluations of three different experimental setups.

5.1. Implementation Details

Pipelines for Deep PNL and LinPNL are implemented using Pytorch. g_i^{-1} in LinPNL and f_i, g_i^{-1} in Deep PNL are all modeled with neural networks with one hidden layer of 20 ReLU units, and an output layer with sigmoid activation. For both Deep PNL and LinPNL, ψ_i's are implemented as neural networks with one hidden layer of 10 hidden ReLU sigmoid units, and an output layer with sigmoid. The choice of nonlinearities may require more investigation. Optimization is done with ADAM, with learning rates tuned for a desirable decrease in the loss. For LinPNL, we add an L1 loss that encourages sparsity of W, which we initialized with 1's for the diagonals. For every update, we zeroed out the gradients of the diagonal elements of W to keep the elements to be 1 throughout training. For Deep PNL, L1 regularizer on all the weights is added to the loss, although we plan to improve upon this naive sparsity constraints in future work.

5.2. Experiment 1: CauseEffectPairs Dataset

The CauseEffectPairs dataset, introduced by Mooij et al. (2016)[1], comprises of multiple two-variable datasets with ground truth causal directions and real feature values, e.g. altitude, temperature. There are total of 108 datasets of pairs each with different causal directions, and 99 were used for the experiment. The goal of this experiment is to see how well LiNGAM, LinPNL, and Deep PNL performs in predicting the true causal directions and compare their accuracies. To interpret the directions predicted by LinPNL, it is required to inspect the matrix W learned and figure out the non-zero off-diagonal component. In our experiments, it was difficult to force one of the off-diagonal elements to be strictly or close to zero while other being non-zero. Therefore we decided to consider the off-diagonal entries with smaller absolute value to be zero, as an approximation. With this prediction scheme for LinPNL, Table 2 shows the accuracy values for LiNGAM and LinPNL, from which we can observe that LinPNL outperforms LiNGAM but with a small margin. However, considering that random guessing will have the accuracy of 50 percent (predicting either one direction or another), their performances do not provide much significance. We can attribute the LiNGAM's failure to the data not following the assumptions for LiNGAM, and LinPNL's failure to premature convergence and approximation error in analyzing the weight matrix W.

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

Model	# Causal Directions Correctly Predicted
LiNGAM	36 / 99
LinPNL	39 / 99

Table 2. Accuracy of causal direction prediction on CauseEffect Pairs dataset. LinPNL outperforms LiNGAM, but not significant.

In order to interpret Deep PNL's results on CauseEffect pairs dataset, not only the inspection of the weights learned, but also the actual function values of $f_i(pa_i)$'s are required. The ideal scenario would be when the sparsity constraint imposed on the loss function forces the weights of certain f_i's to be all zero, which was not the case in our experiment. Some examples of the function values f_i's for all possible values of pa_i's in the dataset is shown in Figure 5. When $f_1(x_2) = 1$ and $f_2(x_1) = 0$ just like the first example in the figure, the predicted direction is $x_2 \to x_1$ because $pa_1 = x_2$. Just like we approximated the results based on the off-diagonal entries of W for LinPNL, when $f_1(x_2) > f_2(x_1)$, roughly the predicted direction from Deep PNL is approximated as $x_2 \to x_1$ and vice-versa. But the experiment did not provide clear directions for most of the datasets, like the last example in the figure.

5.3. Experiment 2: Simulated Dataset

When the underlying data has a linear causal relationship with non-Gaussian noise (i.e. satisfies the LinGAM assumptions (Shimizu et al., 2006)), the LiNGAM model introduced in Section 4.2 should be able to recover the causal directions between the variables. At the same time, such data would also satisfy the assumptions of LinPNL, with the non-linearities (g's) not being a non-linearity, but being either an identity map or a linear function.

Therefore, we simulate data according to the LiNGAM assumptions and test and compare the performance of LiNGAM and LinPNL on this dataset. The data was generated as follows.

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}$$
$$x_i = \sum_j b_i^j pa_i^j + e_i$$
$$b_i^j \sim Unif(0, 10), e_i \sim Unif(0, 0.1)$$

Here, $\mathbf{B}$ is simulated as a lower triangular matrix, so that the parents of a variable x_i are $\{x_j | j < i\}$. Therefore, between any 2 different variables, $x_i, x_j, i \neq j$, there is always a causal relationship between them as either ($x_i \to x_j$) or ($x_j \to x_i$). The number of variables (dimension of $\mathbf{x}$) was varied between 2 and 6, and 2000 datapoints were generated in each trial, for a total of 10 trials. The accuracy

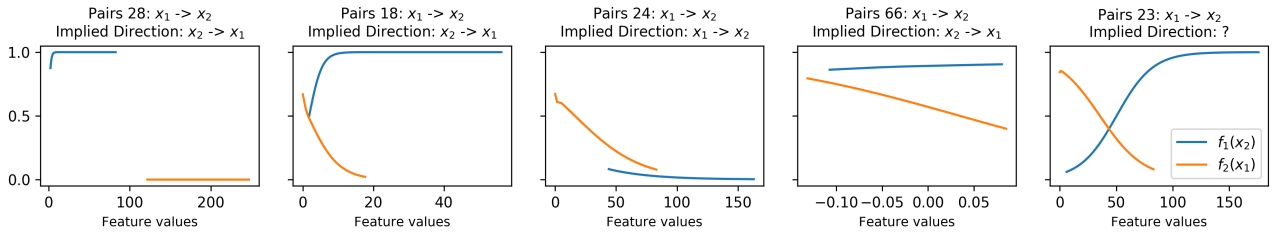

Figure 5. Directions predicted by Deep PNL. By comparing the values of f_i's we can recover some information about the directions learned by Deep PNL, but most of them were inconclusive.

was measured by comparing the mean number of correct causal directions identified across 10 trials.

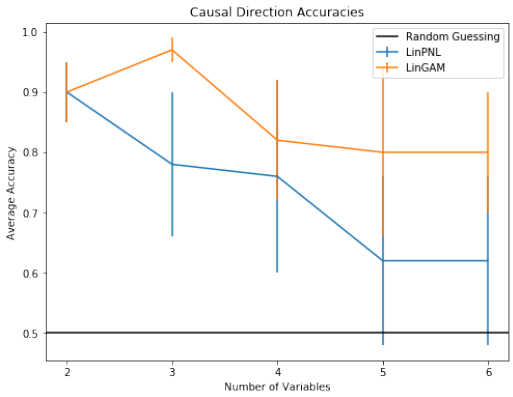

Figure 6. Mean accuracy of LinPNL and LiNGAM in identifying the correct causal relationships for 2 to 6 simulated variables

Figure 6 shows that LiNGAM performs better than LinPNL in identifying the causal relationships. However, it is worth noting that the data was generated exactly according to the LiNGAM assumptions, hence the data was optimized for the LiNGAM model. At the same time, LinPNL does reasonably well to identify the correct causal relationships, and outperforms random guessing by a considerable margin, indicating that LinPNL still is a competitive method for causal discovery with linear causal relations.

5.4. Experiment 3: LASSO Regression Comparison

In practice, techniques such as linear regression with LASSO regularization are frequently used to extract associations. While these associations are not necessarily causal, they provide insight into which features are relatively predictive of a target of interest. The goal of our final experiment is to compare the relationships extracted by causal discovery methods versus those extracted by linear regression with strong regularization. This is particularly relevant for many real world datasets for which we do not know the underlying causal direction/relationship.

Feature	LASSO	LiNGAM	LinPNL
fixed acidity	0.000	effect	cause
volatile acidity	**-0.017**	cause	cause
citric acid	0.000	effect	cause
residual sugar	0.000	effect	cause
chloride	**-0.001**	cause	cause
free SO_2	0.000	effect	cause
total SO_2	**-0.003**	effect	cause
density	0.000	cause	cause
pH	0.000	effect	cause
sulphate	**-0.006**	effect	cause
alcohol	**-0.030**	effect	cause

Table 3. Regression coefficients of LASSO regression and causal directions predicted by LiNGAM and LinPNL.

The dataset for this experiment contains approximately 1,600 samples of Portuguese "Vinho Verde" red wine variants. Features include physiochemical properties (alcohol, acidity, sulphates, etc.) of the wines, and each sample is labeled with a numerical score measure quality. Features are standardized, and the quality score is scaled to be between 0 and 1. For this dataset, an intuitive assumption is that physiochemical properties cause quality and not the other way around. Note, however, that not all physiochemical properties will necessarily have a causal relationship with quality in the first place.

Table 3 contains the regression coefficients and causal directions of the LASSO regression, LinGAM, and LinPNL models. The regularization parameter for LASSO regression was tuned to be $\alpha = 0.005$, which achieved a mean squared error of 0.0046, and the L1 penalty for LinPNL was tuned to be 0.

We observe that while LiNGAM picks up two of the features selected by LASSO, it misses the highly predictive alcohol feature as a cause of quality. In contrast, LinPNL identifies all of the chemicals as causes, which could be correct (though difficult to verify). While this experiment

only gives a partial view of the relationships extracted by causal discovery methods versus predictive methods, it does give some insight into how they differ on real data where assumptions are not perfectly satisfied.

6. Conclusion

In this work, we present Deep PNL and LinPNL for causal discovery, which are variants of PNL and LiNGAM using deep neural networks, with an end-to-end learning scheme using mutual information loss. We evaluate the model on several datasets for causal direction prediction. As evidenced in the experimental section for the LinPNL and Deep PNL models, we sometimes end up with ambiguous models that predict causal directions both ways.

There are two main reasons for this. While the model is flexible enough to represent the causal relationships between many variables, the model keeps getting stuck in bad local minima due to difficulty of minimizing the mutual information objective. This is a well-known problem and is an area of active research. With just a single gradient signal from the loss function of mutual information, the model needs to learn the CDFs used for MISEP, along with the weight matrices for Deep PNL/ LinPNL. We plan to explore different structures of MLPs and nonlinearities that compose each part of the model to improve the optimization. The second is lack of effectiveness of L1 being the regularizer for encoding causal constraints. For example, if the true causal relationships have coefficients 0 and 20, L1 regularization may not quite optimize for the type of sparsity we would like. That is, it is not the size of the coefficients that we care about, but rather how many of them are nonzero. To address this issue in the future, we plan to come up with better regularization constraints, e.g a differentiable variant of L0 regularization similar to (Louizos et al., 2017b).

Moreover, we can study more stable optimization procedure for minimizing joint mutual information, which is a well known research problem (Belghazi et al., 2018). With this, we can then apply the model for causal discovery for larger real world datasets.

References

Almeida, L. B. Misep–linear and nonlinear ica based on mutual information. *Journal of Machine Learning Research*, 4(Dec):1297–1318, 2003.

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

Cho, H., Berger, B., and Peng, J. Reconstructing causal biological networks through active learning. *PloS one*, 11(3):e0150611, 2016.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017a.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017b.

Mooij, J., Janzing, D., Peters, J., and Schölkopf, B. 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*, pp. 745–752. ACM, 2009.

Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., and Schölkopf, B. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

Pearl, J. *Causality*. Cambridge university press, 2009.

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

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

Tran, D., Ranganath, R., and Blei, D. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pp. 5523–5533, 2017.

Zhang, K. and Hyvärinen, A. On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 647–655. AUAI Press, 2009.

Zhang, K. and Hyvärinen, A. Distinguishing causes from effects using nonlinear acyclic causal models. In *Causality: Objectives and Assessment*, pp. 157–164, 2010.

Zhang, K., Wang, Z., Zhang, J., and Schölkopf, B. On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):13, 2016.

Zhang, K., Huang, B., Zhang, J., Glymour, C., and Schölkopf, B. Causal discovery from nonstationary/heterogeneous data: Skeleton estimation and orientation determination. *IJCAI : proceedings of the conference*, 2017:1347–1353, 2017.