Causal Mosaic: Cause-Effect Inference via Nonlinear ICA and Ensemble Method

Pengzhou A. Wu Kenji Fukumizu

The Institute of Statistical Mathematics, Open House, 18 June 2021

Summary

We address the problem of distinguishing cause from effect in bivariate setting.

- Train non-parametric and non-additive causal models on cause-effect pairs, implemented by **neural network**
- Build Causal Mosaic: a causal pair's mechanism is treated as an ensemble mixture of similar mechanisms

Contributions:

1. Two novel cause-effect inference rules with identifiability proofs
2. An ensemble framework that works for real world datasets with only limited labeled pairs
3. A neural network structure designed for causal-effect inference

Problem Setting

We focus on bivariate cases, where there are only two possibilities: either X_1 or X_2 is the direct cause of the other, as shown in the figure:

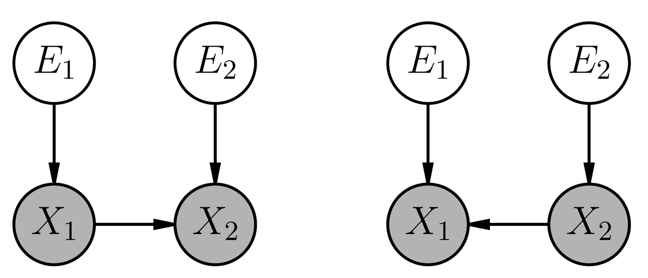

Figure 1. Causal graphs of bivariate SCMs

Their structural causal models (**SCM**s) are the following (1) for $X_1 \to X_2$, and (2) for $X_2 \to X_1$.

$$X_1 = f_1(E_1), \qquad X_2 = f_2(X_1, E_2) \qquad (1)$$
$$X_1 = f_1(X_2, E_1), \qquad X_2 = f_2(E_2) \qquad (2)$$

Intuition

Systems that seem to have different mechanisms can actually share the **same** mechanism.

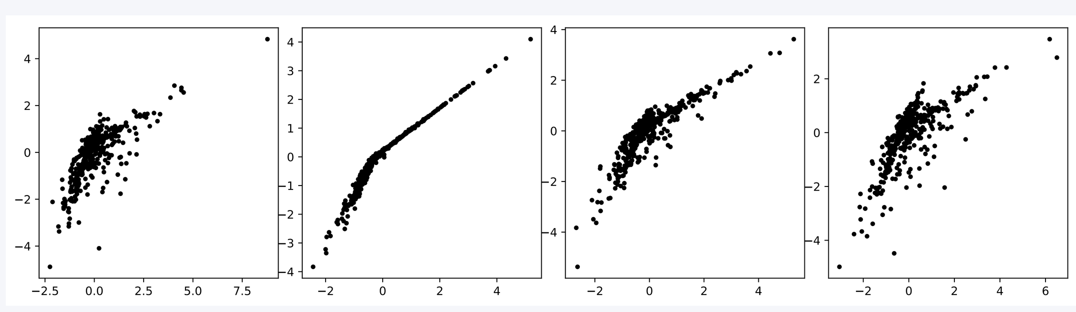

Figure 2. Artificial causal pairs sharing same mechanism.

Learning Shared Mechanism

Theorem 1 (Time-Contrastive Learning (**TCL**) on causal pairs). (informal)

A1. Causal pairs $\mathcal{X}(P) := \{\mathbf{X}_p\}_{p=1}^{P}$, **share SCM** $\mathbf{X}_p = \mathbf{f}(\mathbf{E}_p)$, **exponential family** distribution $p_{E_{i,p}}(e) = \exp[T_i(e)\eta_i(p) - A(\eta_i(p))]$.

A2. Parameters $\{\eta_i(p)\}$ have enough variability.

A3. Train a multilayer perceptron (**MLP**) $\mathbf{h}$, with a final softmax layer to classify all sample points of the pairs, with **pair index used as class label**.

Then, we can **identify** (recover) the **sufficient statistics** by $\mathbf{T}(\mathbf{E}_p) = \mathbf{hICA}(\mathbf{X}_p)$, that is, $\mathbf{h}(\mathbf{X}_p)$ followed by linear ICA.

Separation of Training and Testing

Corollary 1 (**Transferability** of TCL). (informal)

A1 & A2. Training pairs $\mathcal{X}^{tr}(P)$, testing pair $\mathbf{X}^{te}$ with the **same** $\mathbf{f}$ and $\mathbf{T}$ as $\mathcal{X}^{tr}(P)$, **different** η_i.

A3. All possible values of testing pair are seen in training pairs.

A4. Learn $\mathbf{h}$ on $\mathcal{X}^{tr}(P)$ as in A3 of Theorem 1.

Then, we have $\mathbf{T}(\mathbf{E}^{\text{te}}) = \mathbf{hICA}(\mathbf{X}^{\text{te}})$.

Intuitively, after we successfully learned TCL $\mathbf{h}$, we can re-use it to analyze other **unseen** pairs that have the same SCM and sufficient statistics as the training pairs.

Inference Algorithm

Algorithm 1: Inferring causal direction

input : $\sigma(\mathcal{X}^{tr}(P))$, $\sigma(\mathbf{X}^{te})$, $Direction^{tr}$, `align`, `inferule`

output: $Cause^{te}$

Align training set, exploiting $Direction^{tr}$:

$\mathcal{X}^{al}(P) = \mathtt{align}(\sigma(\mathcal{X}^{tr}(P)), Direction^{tr})$

Learn TCL $\mathbf{h}$ on $\mathcal{X}^{al}(P)$

foreach $\alpha = \alpha_0, \alpha_1$ **do**

  $(C_1, C_2)^T_\alpha = \mathbf{hICA}(X^{te}_{\alpha(1)}, X^{te}_{\alpha(2)})$

Run inference rule:

$Cause^{te} = \mathtt{inferule}(\mathbf{C}_{\alpha_0}, \mathbf{C}_{\alpha_1}, \sigma(\mathbf{X}^{\text{te}}))$

Identifiability Result

Theorem 2 (**Identifiability** by independence of hidden components) In Algorithm 1, let:

$Direction^{tr} = \{c_p\}_p^{p=P}$ where $c_p \in \{1, 2\}$ is the cause index: $X^{tr}_{c_p,p} \to X^{tr}_{3-c_p,p}$,

$\mathtt{align} = \{X^{tr}_{c_p,p}, X^{tr}_{3-c_p,p}\}_p^{p=P}$,

$\mathtt{inferule} = \alpha^*(1), \alpha^* = \underset{\alpha \in \{\alpha_0, \alpha_1\}}{\operatorname{argmax}} \mathtt{dindep}(\mathbf{C}_\alpha)$,

`dindep` measures degree of independence.

And assume:

A1. Causal Markov assumption and causal faithfulness assumption hold for data generating SCMs and analysis procedure *except* for a realized nonlinear ICA.

A2. $\mathcal{X}^{tr}(P)$ and $\mathbf{X}^{te}$ satisfy Corollary 1.

Then, the `inferule` defined above (`inferule1` afterwards) identifies the true cause variable.

Asymmetric MLP

Proposition 1 (Inverse of bivariate SCM). For any analyzable SCM as shown in (1), denote the whole system $\mathbf{X} = \mathbf{f}(\mathbf{E})$, if the Jacobian matrix of $\mathbf{f}$ is invertible, then f_1 is invertible.

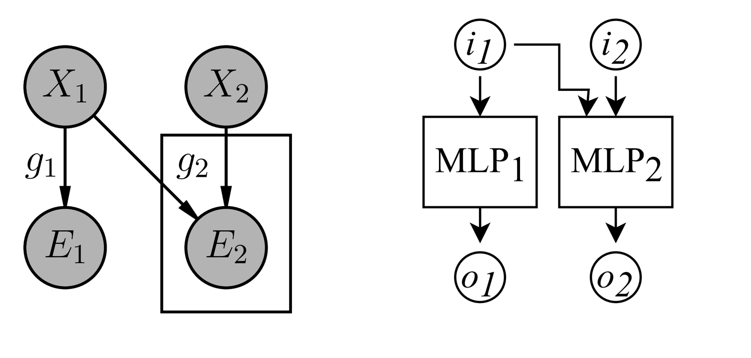

Figure 3. Inverse bivariate analyzable SCM (left) and the indicated MLP structure (right).

Experiments

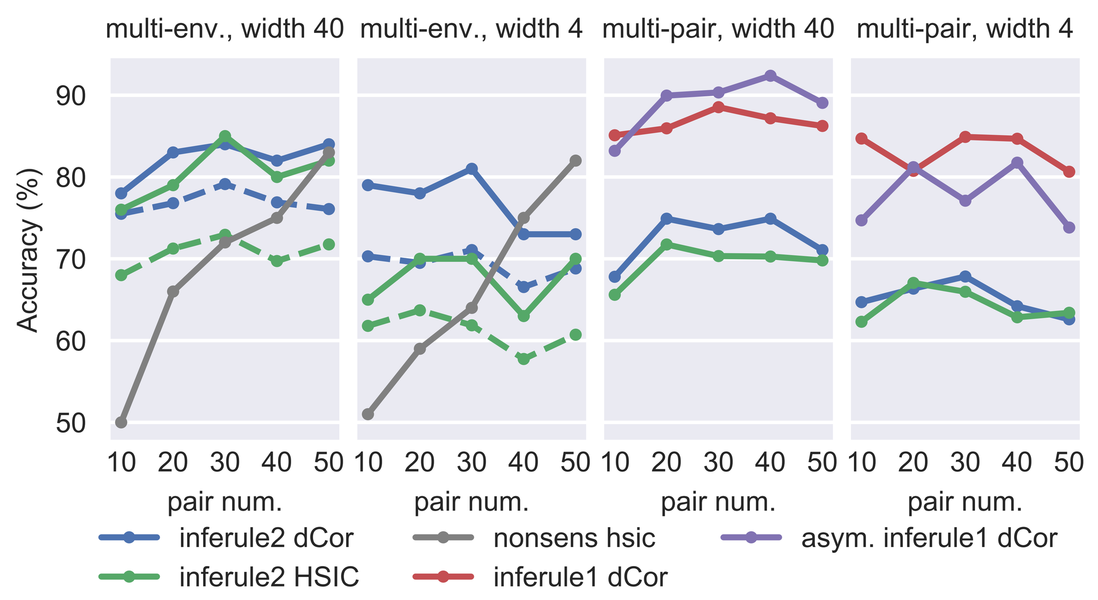

Figure 4. Performance on artificial data.

Table 1. Accuracy (%) on **TCEP** (**real-world** benchmark).

ANM	IGCI	RECI	NCC	OURS
52.5/52.0	60.4/60.8	70.5/62.8	51.8/56.9	$\mathbf{81.5}_{\pm 4.1}/\mathbf{83.3}_{\pm 5.2}$

Take-home Message

"Mosaic" view: real-world causal systems are **diverse**, so we should not fit all the different systems at once; instead, **study at a time a small number** of them that share common aspects, and **then build a whole picture**.

Our Solution

S: the set of all labeled causal pairs we have at hand, c_s: the true cause index for $s \in S$.

Problem of diversity: it is unlikely that most training pairs have same SCM and in same exponential family distribution (A1 of Theorem 1).

Solution:

Random training of TCLs $\{\mathbf{h}_n\}_{n=1}^N$:

1. Train a **large number** (N) of TCLs on sets of randomly chosen **small number** of pairs $\{T_n\}_{n=1}^N$
2. On each T_n, we train MLP M times with randomly chosen hyperparameters

Select TCLs by training and validation accuracy:

1. For each t in T_n, use $\mathbf{hICA}_n$, run line 3–5 of Algorithm 1 on t, get inferred direction $\hat{c}_t$, **training accuracy** $Tacc_n = |\{t : \hat{c}_t = c_t\}|/|T_n|$
2. For each l in $S \setminus T_n$, as step 1., get **validation accuracy** $Vacc_n(l)$ for $\mathbf{h}_n$ on $(S \setminus T_n) \setminus \{l\}$
3. For each s and each n, add n to **selected index set** TSR_s for s, if $s \notin T_n$ **and** $Tacc_n > ThreT$ **and** $Vacc_n(s) > ThreV$

Build a **whole picture** by ensemble method:

1. For each s and each n in TSR_s, infer $Direction_{ns}$ on $\mathbf{h}_n$, by Algorithm 1
2. Calculate weighted prediction $Score_s = \sum_{n \in TSR_s} w_n w_{ns} Direction_{ns}$
3. w_n: how well the training pairs T_n fit together, by the average $\mathtt{dindep}(\mathbf{hICA}_n(.))$ on T_n
4. w_{ns}: pair-specified weight, by the $\mathtt{dindep}(\mathbf{hICA}_n(.))$ for a testing pair s