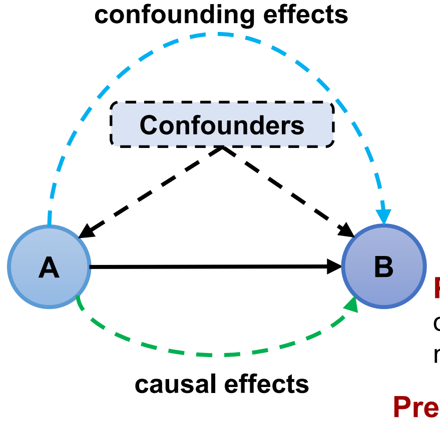
A Generative Adversarial Framework for Bounding Confounded Causal Effects

Yaowei Hu†, Yongkai Wu‡, Lu Zhang†, Xintao Wu†

†University of Arkansas, ‡Clemson University

Background and Motivation

- Inferring causal effects from observational data is an important task in many fields.
- Pearl's structural causal model is a decent and widely adopted framework for conducting causal inference.



- Under the assumption of no hidden confounding, the ACE can be calculated using the well-known truncated factorization formula.

When hidden confounders exist, the ACE may not be uniquely calculated from the observational data without further assumptions, known as the **unidentifiable situation**.

Problem: In an unidentifiable situation, any estimation of ACEs only based on the observational distribution is not guaranteed to be correct.

Previous work on bounding ACE [Balke and Pearl, 1997] is limited to categorical endogenous variables.

Our goal: How to bound ACEs to continuous and possibly high dimensional variables when hidden confounders exist.

Proposed Framework

Our framework: We propose to parameterize the unknown exogenous random variables and structural equations of a causal model using neural networks and implicit generative models.

- Estimate response functions from PA_V to V by neural networks with a certain network structure.
- Use the implicit generative model to generate the distribution for the response-function variable.
- Parameterize the causal model by expressing it with response-function variables.
- Formulate an adversarial learning problem for computing the bounds of the ACE.

Response functions.

- To partition the domain of each exogenous variable into a limited number of equivalent classes, each inducing a distinct functional mapping between endogenous variable. These functional mappings are called the **response functions.**
- Response-function variables $\boldsymbol{r}$ are used to parameterize the causal model.

Coping with continues domain

- For each endogenous variable V, a neural network $v = h_V(paV; \theta_V)$ is used as a universal estimator of response functions from PA_V to V.
- To generate different distributions for θ_V, we adopt the implicit generative model $G_V(\boldsymbol{z}_V)$, which generates data by transforming some random noise $\boldsymbol{z}_V$ to the data via some deterministic function.
- **Definition 3.** For a causal model $\forall V \in \mathbf{V}, v = f_V(\mathrm{pa}_V, u_V)$, its parameterized version is given by

$$\forall V \in \mathbf{V}, v = h_V(\mathrm{pa}_V; G_V(\mathbf{z}_V))$$

where generators $G_V(\mathbf{z}_V)$ contain parameters that are to be learned from data.

- The ACE of A on B is given by $ACE(G;\ a1, a0) = E[B|do(a1)] - E[B|do(a2)]$, obtained by sampling the modified parameterized causal model:

$$a = a'; \forall V \neq A, v = h_V(\mathrm{pa}_V; G_V(\boldsymbol{z}_V))$$

- The generative adversarial learning framework is adopted to ensure generated distribution is close to the observational distribution. The objective function is given by

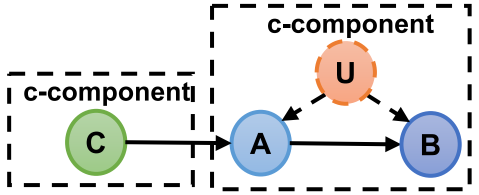
$$V(G;\ D) = E_{\boldsymbol{v} \sim P(\boldsymbol{v})}[\log D(\boldsymbol{v})] + E_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Combining two partial objectives, to obtain the lower bound (similarly for the upper bound), we would like to learn generators G that minimize $ACE(G;\ a_1, a_0)$ subject to that $\max_D V(G, D) \le m + \eta$.
  - m is the theoretical minimal value of $\max_D V(G, D)$.
  - η is a threshold which specifies how close we want the generated distribution to the observational distribution.
- **Problem 1:** Given a causal graph and the data, the lower bound (similarly for the upper bound) of the ACE of A on B is computed by solving the optimization

$$\min_G \max_{\lambda \ge 0} \max_D \{\mathrm{ACE}(G; a_1, a_0) + \lambda (V(G, D) - m - \eta)\}.$$

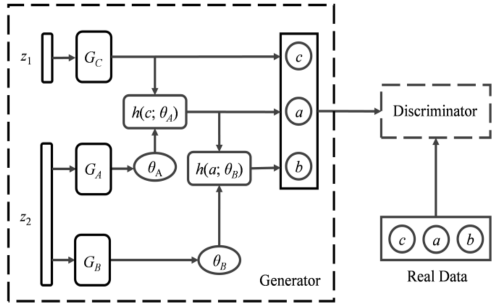
Example

- **Causal graph and equations**



$$c = f_C(\mathbf{u}_C) \qquad c = G_C(\mathbf{z}_1)$$
$$a = f_A(c, \mathbf{u}_A) \xrightarrow{\text{parameterized}} a = h_V(c; G_A(\mathbf{z}_2))$$
$$b = f_B(a, \mathbf{u}_B) \qquad b = h_V(a; G_B(\mathbf{z}_2))$$

- **Architecture of neural networks**

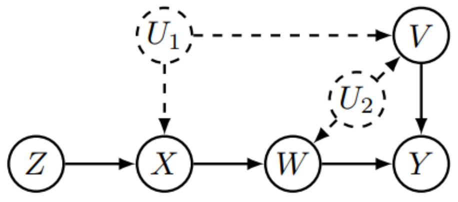


Experiments

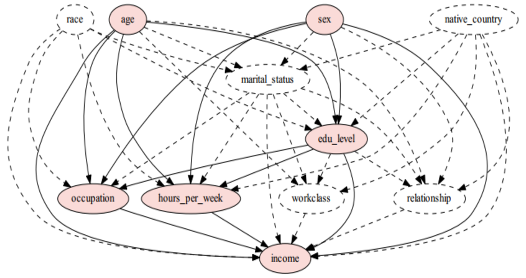
Baselines.

- **Linear/logistic regression**: We build a linear/logistic regression on the outcome using all observed variables, and then compute the ACE based on the coefficient of the treatment variable.
- **Instrumental variable** estimation: We implement this method following the classic instrumental variable formula (Bowden and Turkington 1984).
- **Propensity score** adjusted regression: We adopt the propensity score adjusted regression explained in (Abdia et al. 2017) and follow the method in (Hirano and Imbens 2004) to handle continuous variables.

Datasets.



Causal graph for synthetic data



Causal graph for Adult data

Results.

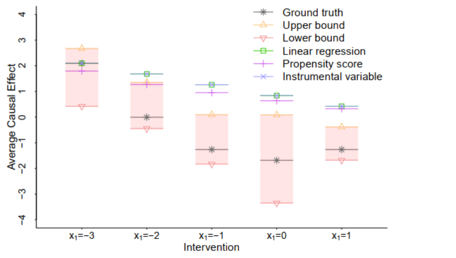


Figure 4: Average causal effects with different interventions ($x_0 = 2$) on the nonlinear synthetic dataset.

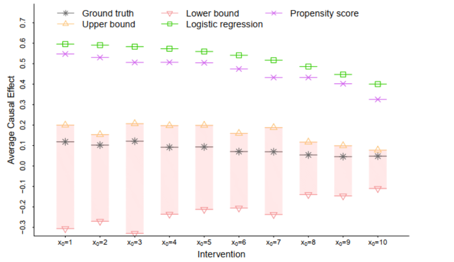


Figure 6: Average causal effects with different interventions ($x_1 = 16$) on the Adult dataset.

Acknowledgement

This work was supported in part by NSF 1646654, 1920920 and 1946391.