CSDS 452 Causality and Machine Learning

Lecture 9: Causal Effect Estimation with Machine Learning / Neural Network

Instructor: Jing Ma

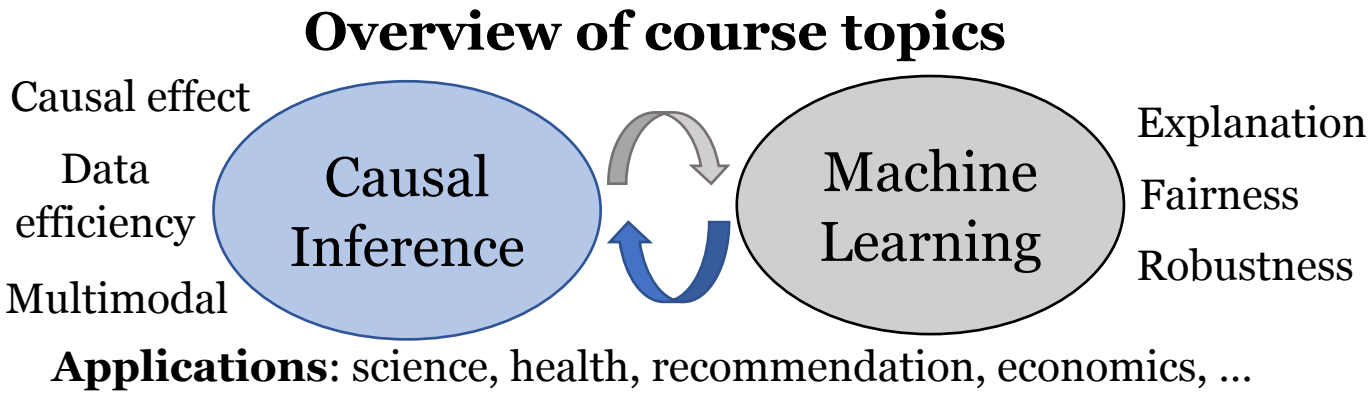
Fall 2024, CDS@CWRU

Outline

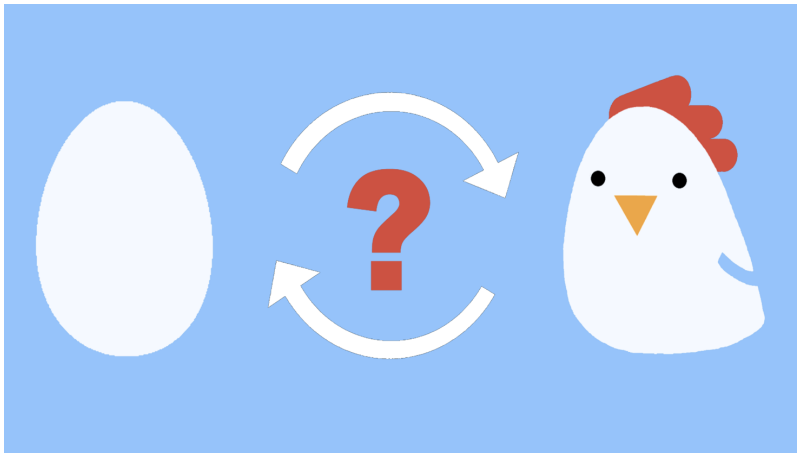
- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

What we have covered

- Three major topics will be covered:
  - ✓ Introduction to causal inference
  - ○ Using machine learning to help causal inference
  - ○ Using causal inference to help machine learning

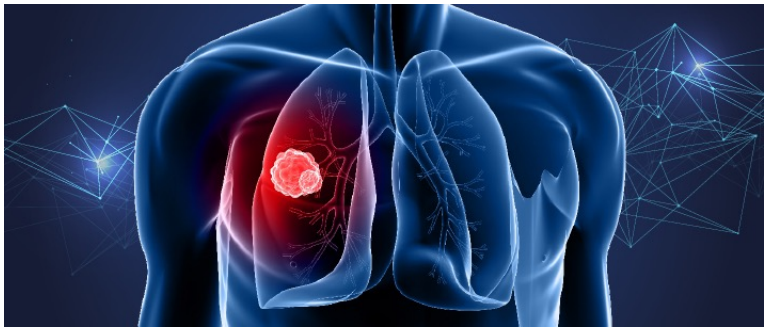


What we have covered



Causal discovery

What causally affects what?



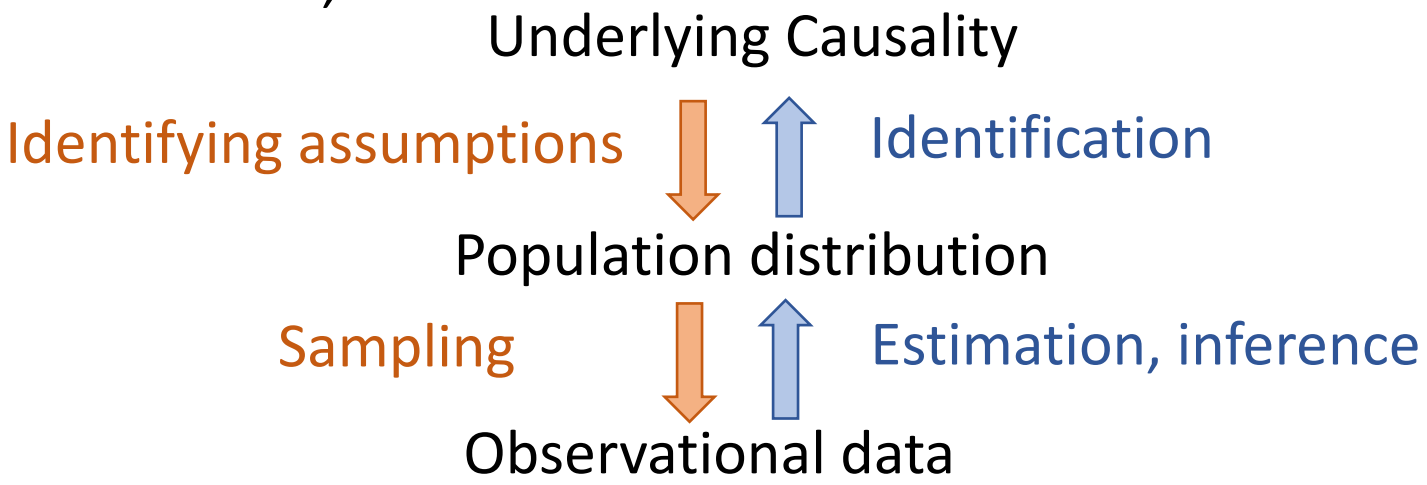
Causal effect estimation How significant is the causal effect?

Recap: Causal Discovery

- Independence-Based Causal Discovery
  - Assumptions:
    - Markov, faithfulness, causal sufficiency, acyclicity
  - Markov Equivalence
  - The PC Algorithm
- Semi-Parametric Causal Discovery
  - No Identifiability Without Parametric Assumptions
  - Linear Non-Gaussian Setting
  - Nonlinear Additive Noise Setting

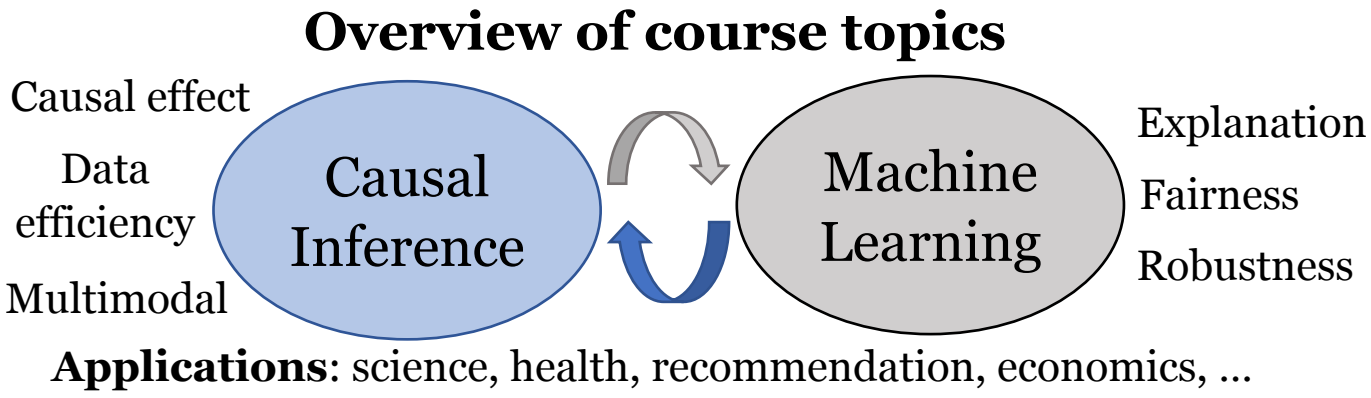
Recap: Causal Effect Estimation

- Identification
  - Identification assumptions (and weaker versions), 3 rules in do-calculus, front-door/backdoor adjustment…
- Estimation:
  - Methods: re-weighting, matching, regression adjustment, difference-in-differences, instrumental variables, …



What will we learn next?

- Three major topics will be covered:
  - ✓ Introduction to causal inference
  - **Using machine learning to help causal inference**
  - Using causal inference to help machine learning



ML for Causal Inference

Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Causal Effect Estimation with Neural Network
  - Counterfactual Regression (CFR)
  - Causal Effect Variational Auto Encoder (CEVAE)

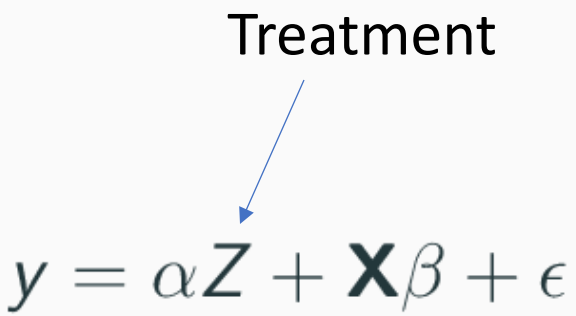
Policy Analytics (moving past the 1900's)

Table 5 ■ Firm value as a function of governance.

Dependent Variable: *Firm q*	(1)	(2)	(3)	(4)	(5)
Property Type q	0.497	0.418	0.403	0.412	0.382
	*(14.33)***	*(7.64)***	*(7.65)***	*(7.16)***	*(10.44)***
EBITDA	0.403	0.456	0.444	0.444	0.163
	*(5.31)***	*(5.17)***	*(5.11)***	*(5.00)***	*(7.21)***
UPREIT	−0.001	−0.006	−0.023	−0.018	
	(0.02)	*(0.09)*	*(0.36)*	*(0.28)*	
Interest Coverage	0.057	0.060	0.043	0.038	−0.004
	(0.74)	*(0.83)*	*(0.63)*	*(0.57)*	*(0.15)*
Mkt Cap	0.127	0.078	0.087	0.096	0.014
	*(2.73)***	*(1.85)**	*(1.96)**	*(2.11)**	*(0.39)*
Excess Comp		−0.002	0.000	−0.002	−0.020
		(0.03)	*(0.01)*	*(0.05)*	*(0.85)*
Instl Ownership		0.053	0.078	0.085	0.101
		(1.00)	*(1.48)*	*(1.50)*	*(2.55)**
Block Ownership			−0.046	−0.041	0.013
			(1.38)	*(1.23)*	*(0.59)*
D&O Ownership			0.106	0.105	0.072
			(1.57)	*(1.55)*	*(2.08)**
Ln(Board Size)				−0.044	−0.097
				(0.77)	*(2.86)****
Outside Board				0.029	0.021
				(0.75)	*(0.93)*
Maryland				−0.026	
				(0.53)	
Fixed Effects?	No	No	No	No	Yes
Observations	882	882	882	882	882
R^2	0.53	0.55	0.56	0.56	0.60
p value from *F* test of null that all governance coefficients are zero		0.61	0.21	0.50	0.00***

Moving from this...

Linear case



... to that!

A more general setting

$$y = f(Z, \mathbf{X}) + \epsilon$$

May be nonlinear

... to that!

$$y = f(Z, \mathbf{X}) + \epsilon$$

Today:

- a general, "default" framework
- a rich output that allows you to ask lots of different questions

Our setting

We'll assume:

- **Observational and experimental data**
- **Conditional unconfoundedness/ignorability** (we've measured all the factors causally influencing treatment and response),
- **Covariate-dependent treatment effects** (individuals can have different responses to treatment according to their covariates) (heterogeneous treatment effect)
- **Binary treatments**

Our assumptions, more formally

Strong ignorability:

$$Y_i(0), Y_i(1) \perp\!\!\!\perp Z_i \mid \mathbf{X}_i = \mathbf{x}_i,$$

Positivity:

$$0 < \Pr(Z_i = 1 \mid \mathbf{X}_i = \mathbf{x}_i) < 1$$

for all i. Therefore

$$\mathrm{E}(Y_i(z) \mid \mathbf{x}_i) = \mathrm{E}(Y_i \mid \mathbf{x}_i, Z_i = z),$$

so the conditional average treatment effect (CATE) is

$$\begin{aligned} \tau(\mathbf{x}_i) :&= \mathrm{E}(Y_i(1) - Y_i(0) \mid \mathbf{x}_i) \\ &= \mathrm{E}(Y_i \mid \mathbf{x}_i, Z_i = 1) - \mathrm{E}(Y_i \mid \mathbf{x}_i, Z_i = 0). \end{aligned}$$

Modeling assumptions

We write

$$\mathrm{E}(Y_i \mid \mathbf{x}_i, z_i) = f(\mathbf{x}_i, z_i),$$

so that

$$\tau(\mathbf{x}_i) := f(\mathbf{x}_i, 1) - f(\mathbf{x}_i, 0).$$

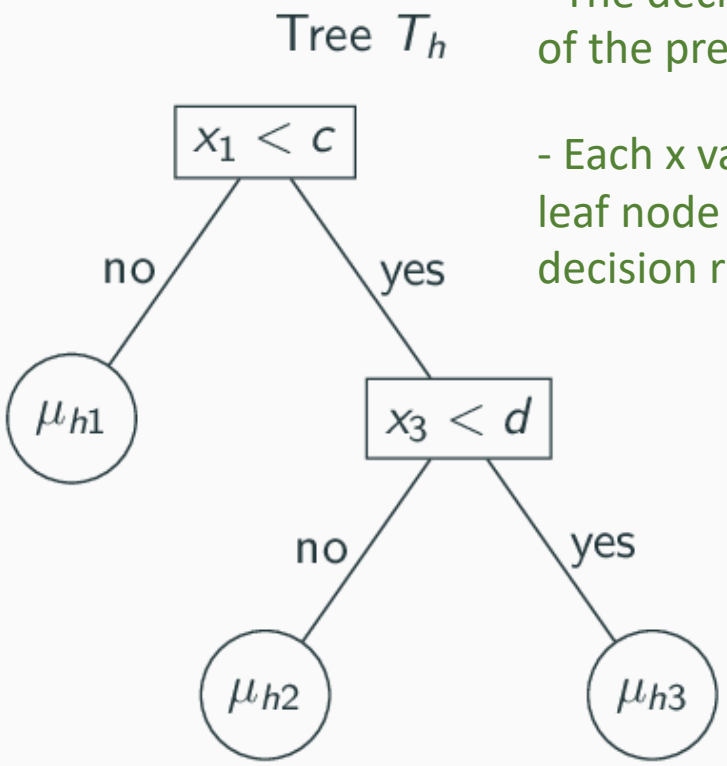
We assume iid Gaussian errors:

$$Y_i = f(\mathbf{x}_i, z_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

nb: Strong ignorability means $\epsilon_i \perp\!\!\!\perp Z_i \mid \mathbf{x}_i$.

How do we regularize estimates of f? (What prior on f?)

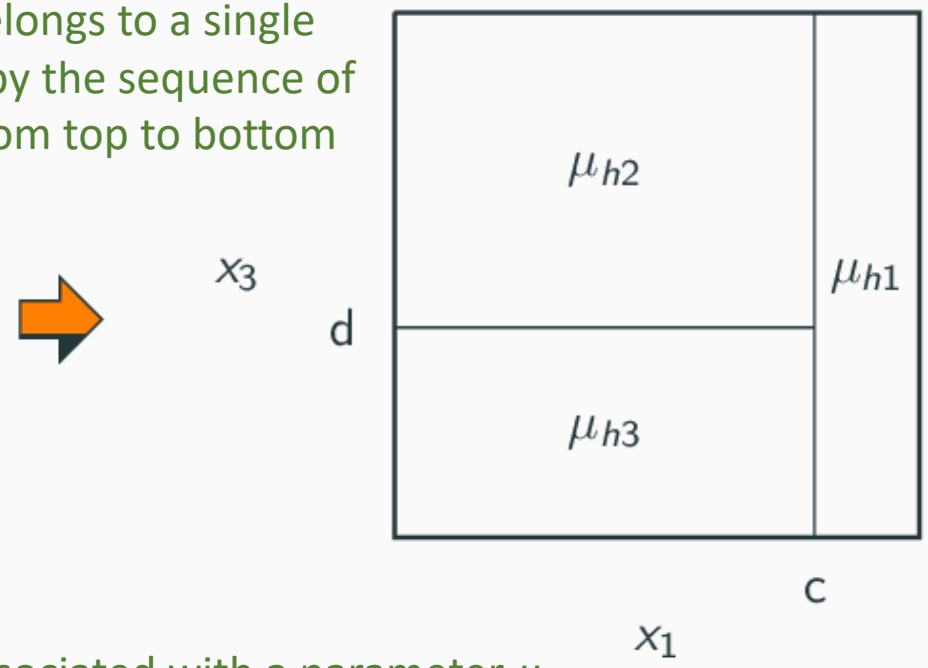
Regression Trees



- The decision rules are binary splits of the predictor space

- Each x value belongs to a single leaf node of T_h by the sequence of decision rules from top to bottom

$g(\mathbf{x}, T_h, M_h)$



Each partition is associated with a parameter μ

Leaf/End node parameters

$$M_h = (\mu_{h1}, \mu_{h2}, \mu_{h3})$$

Partition $\mathcal{A}_h = \{\mathcal{A}_{h1}, \mathcal{A}_{h2}, \mathcal{A}_{h3}\}$

$$g(\mathbf{x}, T_h, M_h) = \mu_{ht} \text{ if } \mathbf{x} \in \mathcal{A}_{ht} \text{ (for } 1 \leq t \leq b_h).$$

Number of partitions

One single tree may be a coarse model, how can we do better?

How about an ensemble of trees?

Bayesian Additive Regression Trees (BART)

Bayesian additive regression trees (Chipman, George, & McCulloch, 2008):

BART has been very successful in prediction

$$y_i = f(\mathbf{x}_i, z_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

$$f(\mathbf{x}, z) = \sum_{h=1}^{m} g(\mathbf{x}, z, T_h, M_h)$$

m: Number of trees

a sum over trees

Hill (2011) proposes adopting Bayesian additive regression trees (BART) for causal inference.

Bayesian Additive Regression Trees (BART)

Bayesian additive regression trees (Chipman, George, & McCulloch, 2008):

BART has been very successful in prediction

$$y_i = f(\mathbf{x}_i, z_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

$$f(\mathbf{x}, z) = \sum_{h=1}^{m} g(\mathbf{x}, z, T_h, M_h)$$

m: Number of trees

a sum over trees

treatment

Hill (2011) proposes adopting Bayesian additive regression trees (BART) for causal inference.

Making BART great for causal inference

BART is great as a prior over regression functions! It also works great for causal inference, but in some settings it can be problematic:

1. With strong confounding, estimates of individual/average treatment effects from BART can exhibit severe bias.

2. With homoegenous effects/moderate heterogeneity, BART's treatment effect estimates are highly variable.

We can fix both of these!

Problem 1: Strong confouding can lead to high bias

Suppose that:

- Y: measure of heart distress,
- Z: took heart medication,
- x_1 and x_2 are blood pressure measurements.

Let's make this easy: $p = 2$, $n = 1,000$, with homogeneous treatment effects ($\tau = 1$).

Problem 1: Strong confouding can lead to high bias

Assume the true model:

$$
\begin{aligned}
Y_i &= \mu(\mathbf{x}_i) - Z_i + \epsilon_i, \\
\mu(\mathbf{x}_i) &= 1 \text{ if } x_{i1} < x_{i2}, \ -1 \text{ otherwise.} \\
\Pr(Z_i = 1 \mid x_{i1}, x_{i2}) &= \Phi(\mu(\mathbf{x}_i)), \\
\epsilon_i \overset{iid}{\sim} \mathrm{N}(0, 0.7^2), &\quad x_{i1}, x_{i2} \overset{iid}{\sim} \mathrm{N}(0, 1).
\end{aligned}
$$

This example demonstrates targeted selection into treatment: Patients with $x_{i1} < x_{i2}$ are 5 times as likely to receive the new drug precisely because they are more likely to have higher levels of heart distress.

Despite low noise, low dimension, and homogeneous effects, BART has problems...

BART is badly biased here

Across 250 simulated datasets with $n = 1000$, BART is badly biased:

Prior	Bias	Coverage	RMSE
BART	0.14	31%	0.15

This is due to a pheonomenon called **regularization induced confounding** (see Hahn, Carvalho, Puelz and He, 2018).

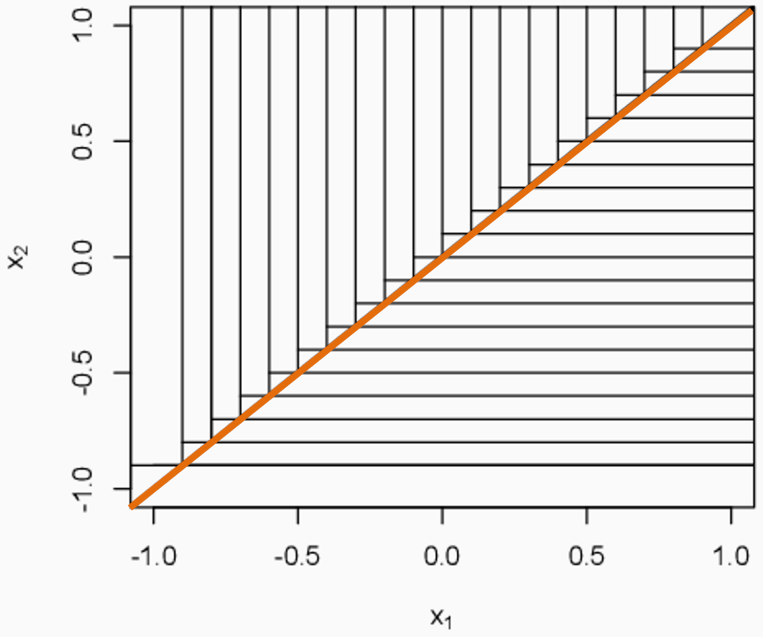
Targeted selection induces regularization induced confounding

Why is BART biased in this example?

- $\pi(\mathbf{x}) = \Pr(Z = 1 \mid \mathbf{x})$ is a noisy function of $\mu(\mathbf{x})$, so $\mu(\mathbf{x})$ "looks like" $\pi(\mathbf{x})$, and $\mu(\mathbf{x})$ is hard to approximate with trees

A regression tree needs to make many splits along the ground-truth function shape of $\mu(x)$

$$
\begin{aligned}
Y_i &= \mu(\mathbf{x}_i) - Z_i + \epsilon_i, \\
\mu(\mathbf{x}_i) &= 1 \text{ if } x_{i1} < x_{i2}, \ -1 \text{ otherwise.} \\
\Pr(Z_i = 1 \mid x_{i1}, x_{i2}) &= \Phi(\mu(\mathbf{x}_i)), \\
\epsilon_i &\overset{iid}{\sim} \mathrm{N}(0, 0.7^2), \quad x_{i1}, x_{i2} \overset{iid}{\sim} \mathrm{N}(0, 1).
\end{aligned}
$$



Targeted selection induces regularization induced confounding

Why is BART biased in this example?

As Z is similar to $\mu(x)$, BART can prefer apportioning the effect to Z because it is simpler than making many splits on x_1, x_2.

- $\pi(\mathbf{x}) = \Pr(Z = 1 \mid \mathbf{x})$ is a noisy function of $\mu(\mathbf{x})$, so $\mu(\mathbf{x})$ "looks like" $\pi(\mathbf{x})$, and $\mu(\mathbf{x})$ is hard to approximate with trees

- Strong confounding means $Z \approx \pi(\mathbf{x})$, and targeted selection means $\mu(\mathbf{x})$ is a function of $\pi(\mathbf{x})$, so $\mu(\mathbf{x})$ can be approximated by a tree that splits on Z

- The BART prior over f penalizes the total number of splits, so to fit $\mu(\mathbf{x})$ BART would rather split on Z once than x_1 and x_2 many times – confusing confounding for treatment effects: **regularization induced confounding** (Hahn et al (2016))

A fix: Propensity Score BART

We can fix this by estimating $\pi(\mathbf{x}) = \Pr(Z = 1 \mid \mathbf{x})$ (here using BART) and including $\hat{\pi}(\mathbf{x})$ as an extra predictor variable

This extra covariate dimension can help BART separate what comes from X and what comes from Z

Prior	Bias	Coverage	RMSE
BART	0.14	31%	0.15
Oracle BART	0.00	98%	0.05
ps-BART	0.06	85%	0.08

(With an ensemble estimate of $\hat{\pi}$, ps-BART$\approx$Oracle BART)

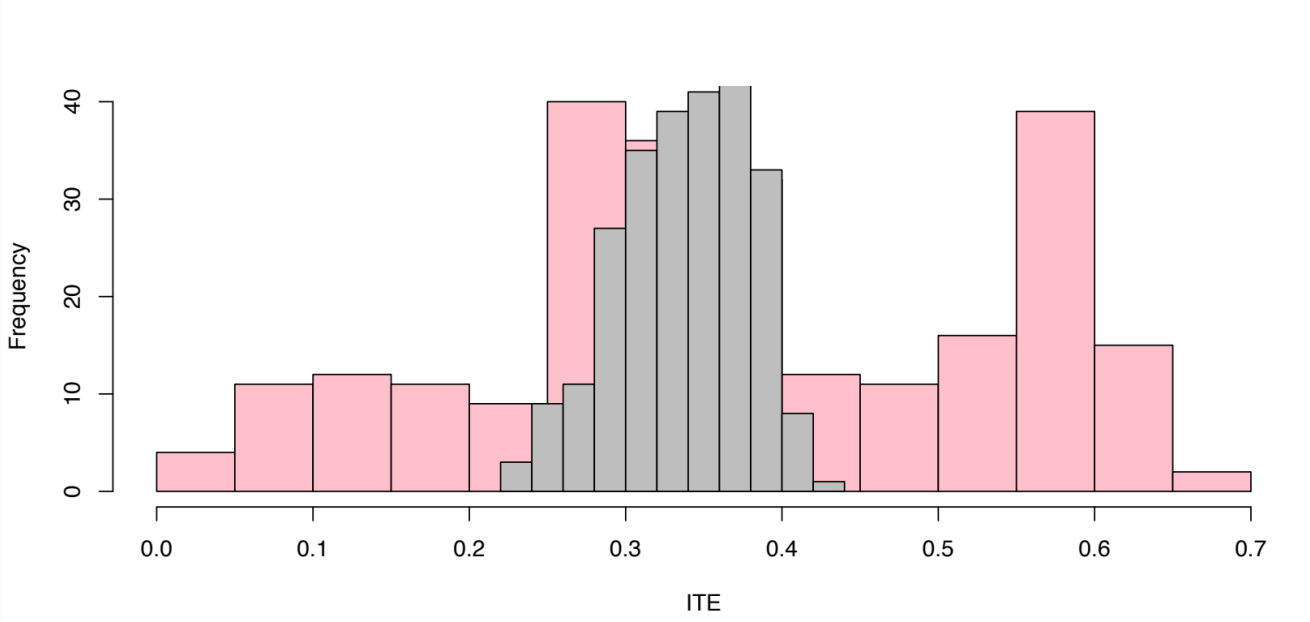
Problem 2: Naive priors give high variance estimates

In the model

This function does not directly focus on the treatment effect of interest

$$y_i = f(\mathbf{x}_i, \hat{\pi}(\mathbf{x}_i), z_i) + \epsilon_i$$

a BART prior on f provides no direct mechanism to regularize the treatment effect function $\tau(\mathbf{x})$



ps-BART is in pink; our fix is in grey

The fix: Bayesian causal forests

Reparameterize!

A "surface" that how X influence Y

How treatment influence outcome (what we are interested in)

$$f(\mathbf{x}_i, z_i) = \mu(\mathbf{x}_i, \hat{\pi}(\mathbf{x}_i)) + \tau(\mathbf{x}_i) z_i,$$

where μ and τ have independent BART priors.

This reparameterization let us focus more on the treatment effect of interest

Now the treatment effect is

$$\tau(\mathbf{x}_i)$$

and we can "shrink towards homogeneity" with stronger regularization on τ, independent of regularization on μ

Tweaking priors in BCF

Several adjustments to the BART prior on τ:

- Higher probability on smaller τ trees (than BART defaults)

Encourage homogeneous treatment effect until there is obvious evidence against it

- Higher probability on "stumps" (all stumps = homogeneous effects)

- $N^+(0, v)$ Hyperprior on the scale of leaf parameters in τ

Robustness

RIC in the wild: 2017 ACIC Data Analysis Challenge

Treatment-response pairs were simulated according to 32 distinct data generating processes (DGPs), given fixed covariates ($n = 4,302$, $p = 58$) from an empirical study.

We varied three parameters among two levels

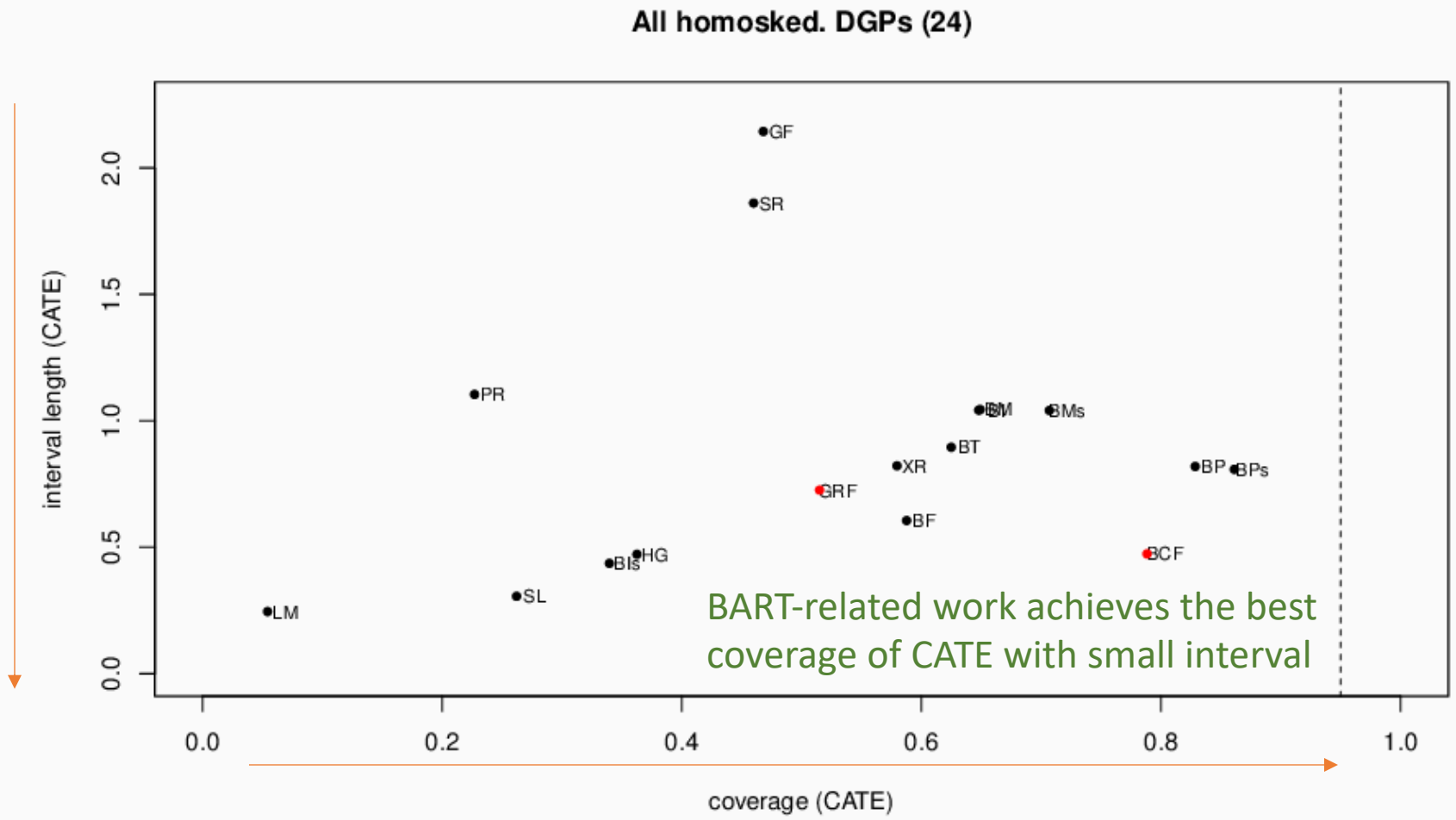
- **High** or **Low** *noise level*,
- **Strong** or **Weak** *confounding*,
- **Small** or **Large** *effect size*.

The error distributions were one of four types

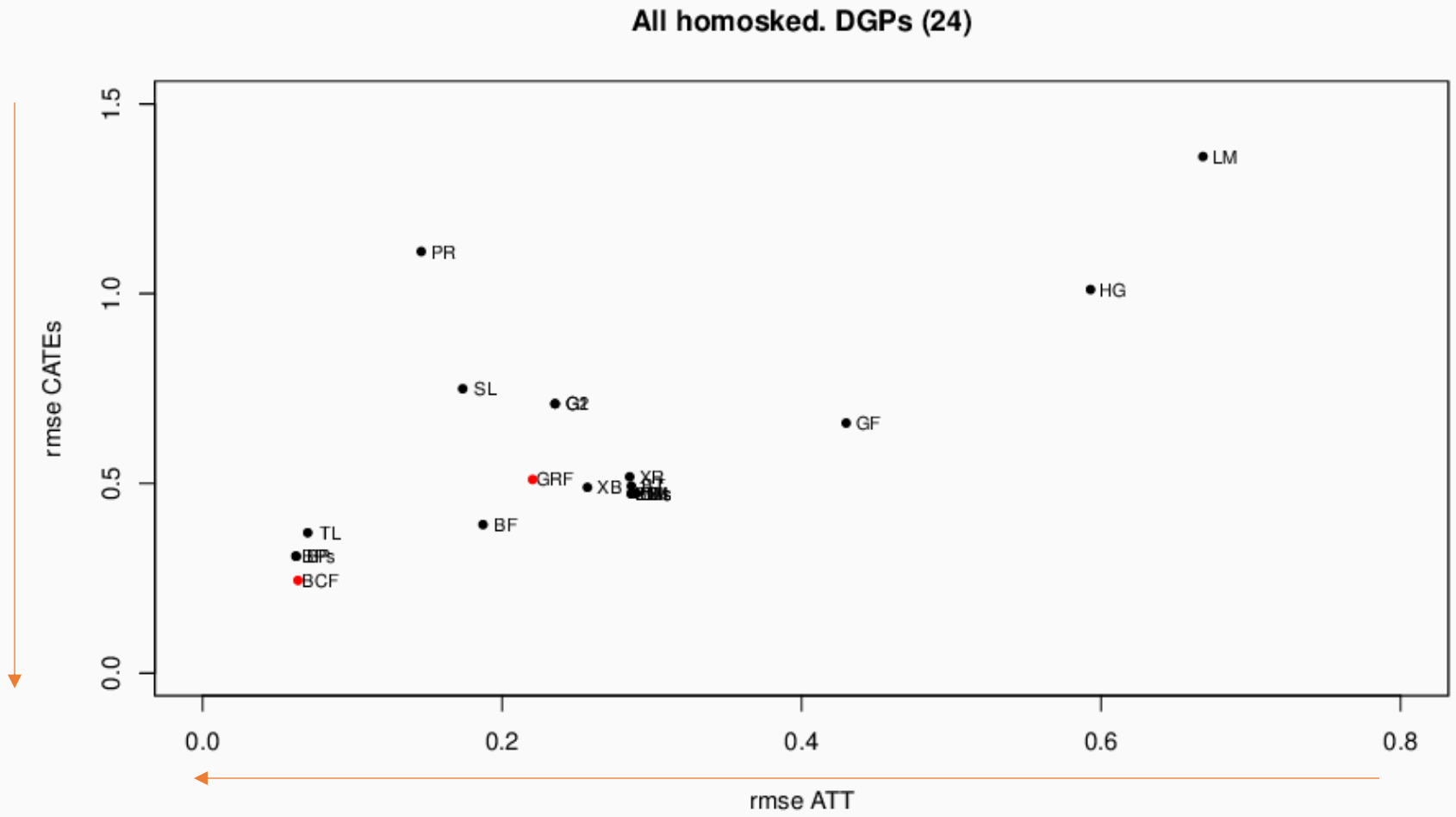
- Additive, homoskedastic, independent,
- Nonadditive, homoskedastic, independent,
- Additive, heteroskedastic, independent.

To assess coverage, 250 replicate data sets were generated for each DGP.

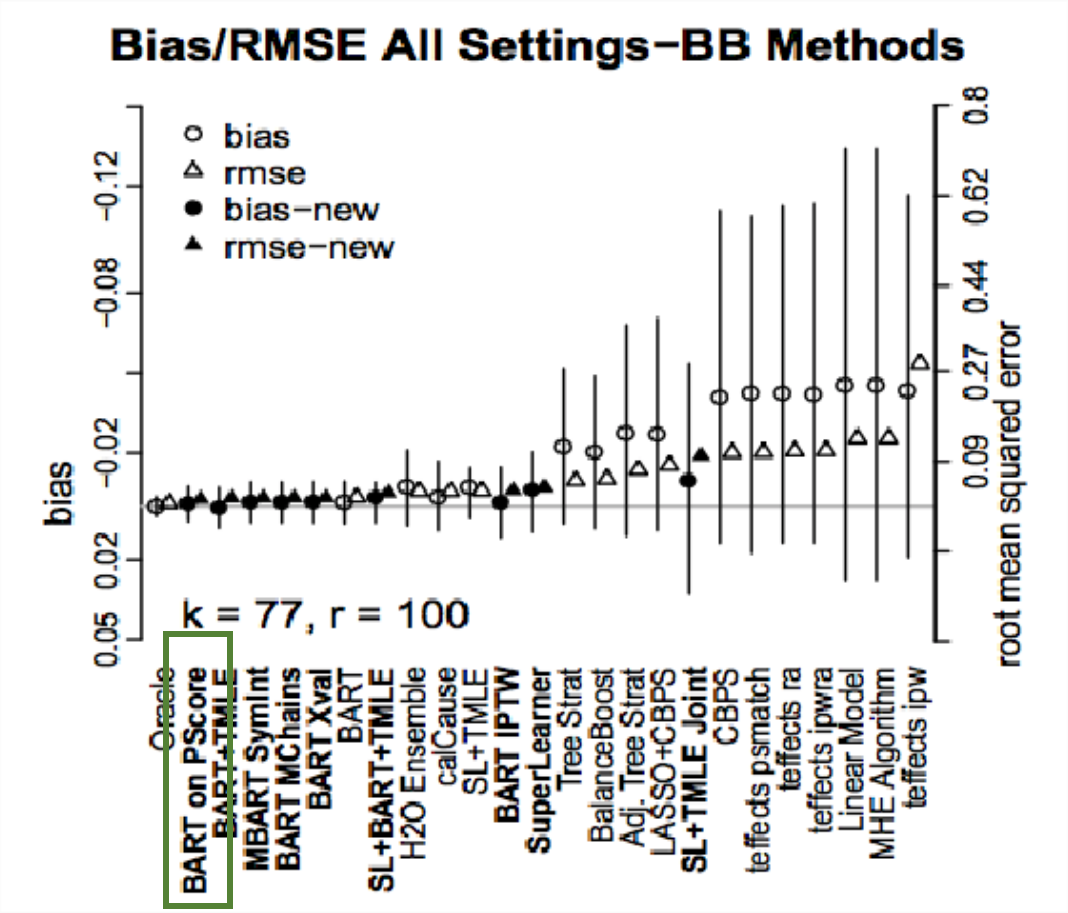
Results: Inference for CATE on homoskedastic DGPs



Results: Estimation for homoskedastic DGPs



ACIC 2016 Redux



Large, highly variable treatment effects and no explicit targeted selection!

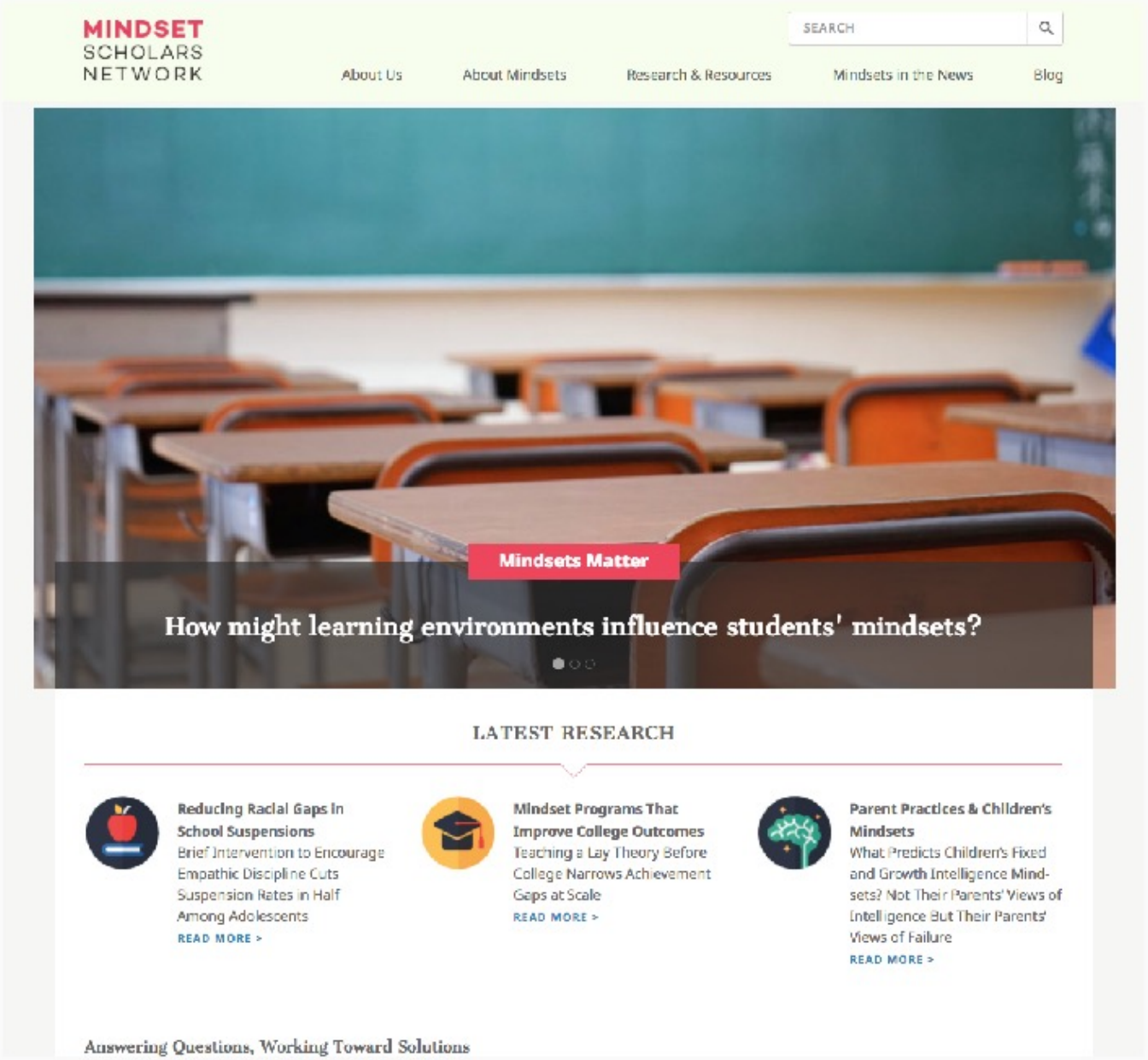
Takeaways

In observational data regularization-induced confounding can adversely bias treatment effect estimates **from any method that uses regularization**. Explicitly modeling selection is necessary for robust inference.

BART is an impressive response surface method for causal inference; new BCF models improve on "vanilla" BART in key respects:

- Propensity score estimates as covariates mitigate RIC
- Reparameterization allows regularization to be imposed robustly and directly on the estimand of interest.
- **It also facilitates extensions to multilevel models!**

Multilevel BCF: The National Study of Learning Mindsets



National Study of Learning Mindsets

- National Study of Learning Mindsets (Yeager et. al., 2017): Randomized controlled trial of a low-cost mindset intervention
- Probability sample of 76 schools ($\approx 14,000$ students)
- Specifically designed to assess treatment effect heterogeneity
- Preregistration plan included specific subgroups of interest, **and a blinded exploratory analysis of heterogeneity**

National Study of Learning Mindsets

Desiderata for our analysis:

- Avoid model selection/specification search
- School-level effect heterogeneity explained and unexplained by covariates
- Interpretable model summaries for communicating results

Multilevel Linear Models for Heterogeneous Treatment Effects

School-specific intercepts/fixed/random effects

School-specific "unexplained" heterogeneity

$$y_{ij} = \alpha_j + \sum_{h=1}^{p} \beta_h x_{ijh} + \left[\sum_{\ell=1}^{k} \tau_\ell w_{ij\ell} + \gamma_j \right] z_{ij} + \epsilon_{ij}$$

i: student
j: school

Controls at the student and/or school level

Moderators at the student and/or school level

Coloring outside the lines: Multilevel Bayesian Causal Forests

We replace linear terms with Bayesian additive regression trees (BART)

$$y_{ij} = \alpha_j + \beta(\mathbf{x}_{ij}) + [\tau(\mathbf{w}_{ij}) + \gamma_j]\, z_{ij} + \epsilon_{ij}$$

Coloring outside the lines: Multilevel Bayesian Causal Forests

We replace linear terms with Bayesian additive regression trees (BART)

BART in causal inferece: Hill (2011), Green & Kern (2012), …

Parameterizing treatment effect heterogeneity with BART is due to Hahn, Murray and Carvalho (2017)

$$y_{ij} = \alpha_j + \beta(\mathbf{x}_{ij}) + \left[\tau(\mathbf{w}_{ij}) + \gamma_j\right] z_{ij} + \epsilon_{ij}$$

Coloring outside the lines: Multilevel Bayesian Causal Forests

We replace linear terms with Bayesian additive regression trees (BART)

BART in causal inferece: Hill (2011), Green & Kern (2012), …

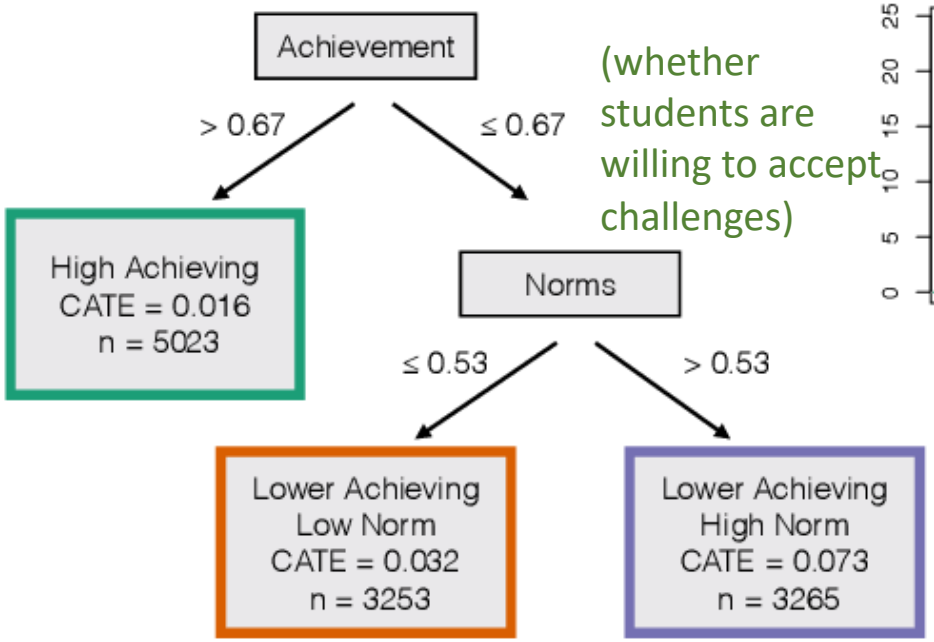
Parameterizing treatment effect heterogeneity with BART is due to Hahn, Murray and Carvalho (2017)

$$y_{ij} = \alpha_j + \beta(\mathbf{x}_{ij}) + \left[\tau(\mathbf{w}_{ij}) + \gamma_j\right] z_{ij} + \epsilon_{ij}$$

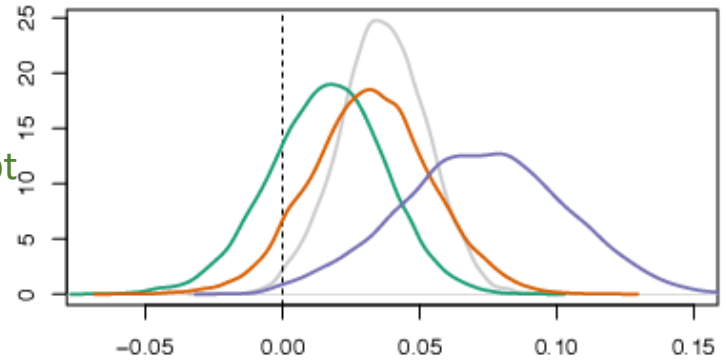
Allows for complicated functional forms (nonlinearity, interactions, etc) without pre-specification…

…while carefully regularizing estimates with prior distributions (shrinkage toward additive structure and discouraging implausibly large treatment effects)

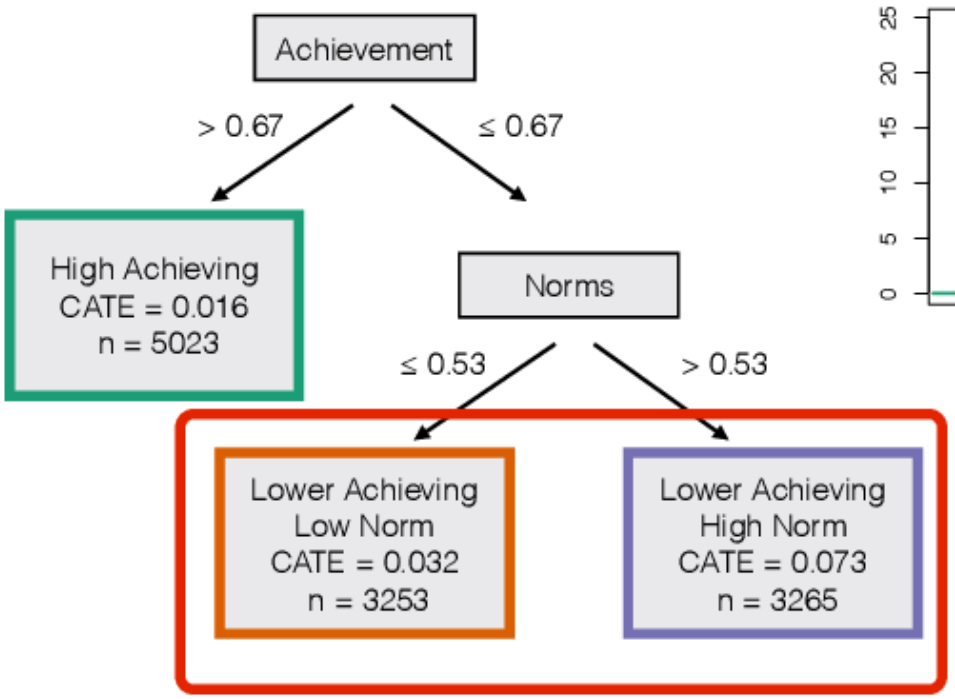
(School achievement)

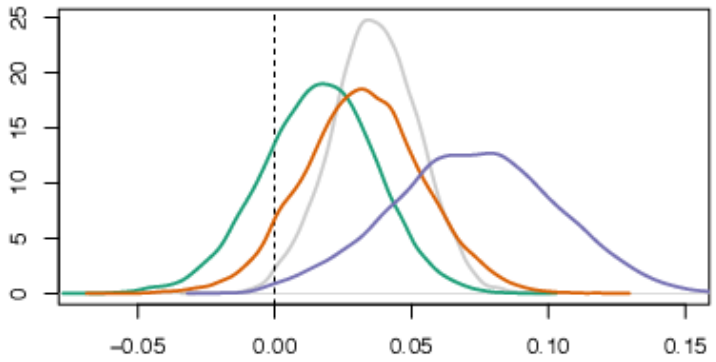


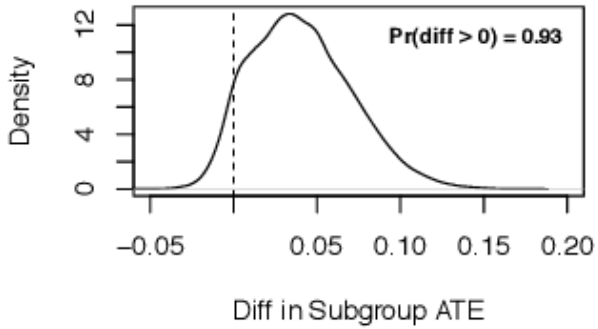
(whether students are willing to accept challenges)



- CATE is low for students in schools which are already successful.
- For other schools, CATE is higher for students who are willing to take challenges.


Achievement
> 0.67
≤ 0.67
High Achieving
CATE = 0.016
n = 5023
Norms
≤ 0.53
> 0.53
Lower Achieving
Low Norm
CATE = 0.032
n = 3253
Lower Achieving
High Norm
CATE = 0.073
n = 3265


0
5
10
15
20
25
−0.05
0.00
0.05
0.10
0.15


Density
0
4
8
12
Pr(diff > 0) = 0.93
−0.05
0.05
0.10
0.15
0.20
Diff in Subgroup ATE

Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

Original Paper:

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

[Johansson et al., 2016] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." International conference on machine learning. PMLR, 2016.

Treatment Effects

In the following, we aim for unbiased estimation of ITE (and ATT)

Individualized Treatment Effect

counterfactual outcome

$$\hat{\mathbf{ITE}}(x_i) = \begin{cases} y_i^F - \hat{y}_i^{CF} & (t_i = 1) \\ \hat{y}_i^{CF} - y_i^F & (t_i = 0) \end{cases}$$

covariates

factual outcome

treated or not

$$h(x_i, 1 - t_i)$$

Counterfactual outcomes are estimated by an arbitrary machine learning model ***h***

Features {covariates x_i , t} -> y

Average Treatment effect on the Treated

$$\hat{\mathbf{ATT}} = \sum_{i \in N_t} \hat{\mathbf{ITE}}(x_i)$$

Problem : covariate shift

Estimating y^{CF} is usually difficult.
The main reason is due to differences in the distribution of F and CF

empirical *factual/counterfactual* distribution

$$\hat{P}^F = \{(x_i, t_i)\}_{i=1}^n$$

not eqaul (if not RCT)

$$\hat{P}^{CF} = \{(x_i, 1 - t_i)\}_{i=1}^n$$

Covariate shift

"the problem of causal inference by counterfactual prediction might require inference over a different distribution than the one from which samples are given. In machine learning terms, this means that the feature distribution of the test set differs from that of the train set. This is a case of covariate shift, which is a special case of **domain adaptation**"

[Johansson et al., 2016]

CFR's Solution

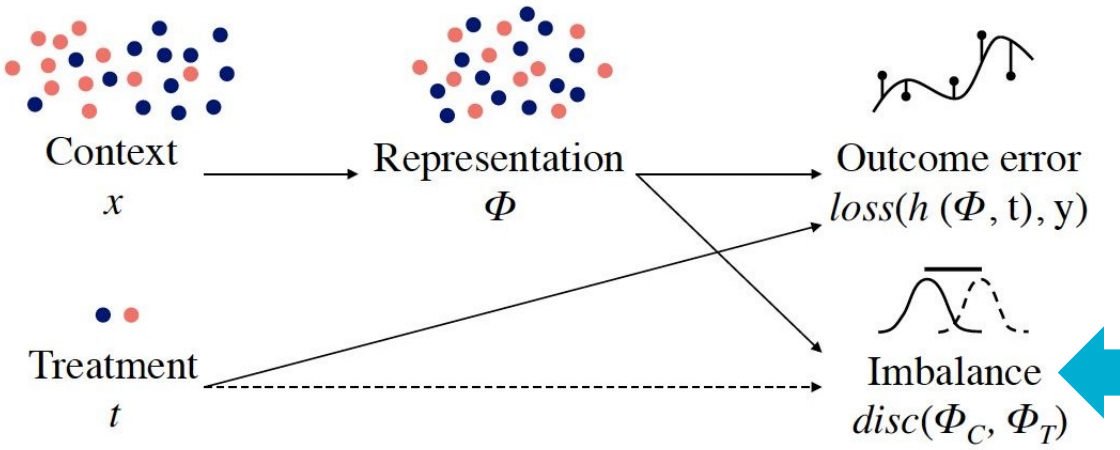


Figure 1. Contexts x are representated by $\Phi(x)$, which are used, with group indicator t, to predict the response y while minimizing the imbalance in distributions measured by $\mathrm{disc}(\Phi_C, \Phi_T)$.

New Loss

-Pseudo-distance between the distributions of both groups in the Representation layer

-By adding this to loss, we design the distribution of both groups to be close in the layer

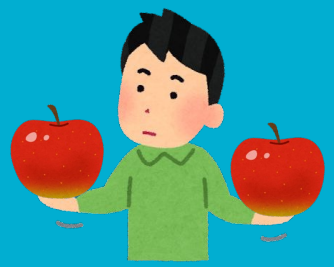


Fig by. [Johansson et al., 2016] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." International conference on machine learning. PMLR, 2016.

Two architectures

- **[Shalit et al., 2017] CFR** framework splits the net to outcomes.
- Balancing neural network (**BNN**) [Johansson et al., 2016] may underestimate the treatment effect (shrinkage estimation) in the outcome net due to regularization bias.

[Johansson et al., 2016]

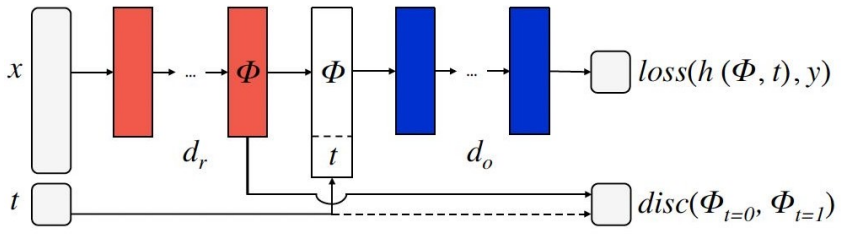


Figure 2. Neural network architecture.

[Shalit et al., 2017]

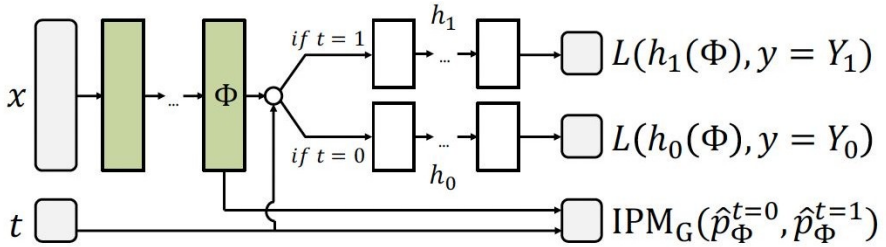


Figure 1. Neural network architecture for ITE estimation. L is a loss function, IPM_G is an integral probability metric. Note that only one of h_0 and h_1 is updated for each sample during training.

[Johansson et al., 2016] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." International conference on machine learning. PMLR, 2016.

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Object function

- Solving the mixed loss minimization problem with outcome losses and pseudo-distances
- **α is a hyperparameter (α>0).**
- If α=0, Treatment-Agnostic Representation Network (**TARNet**)

Sample weighting according to percentage of treatment

L2 penaltiy

$$\min_{\substack{h,\Phi \\ \|\Phi\|=1}} \frac{1}{n}\sum_{i=1}^{n} w_i \cdot L\left(h(\Phi(x_i), t_i), y_i\right) + \lambda \cdot \Re(h)$$

$$+ \alpha \, \mathrm{IPM}_G\left(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\right),$$

$$\text{with } w_i = \frac{t_i}{2u} + \frac{1-t_i}{2(1-u)}, \quad \text{where } u = \frac{1}{n}\sum_{i=1}^{n} t_i,$$

and $\Re$ is a model complexity term.

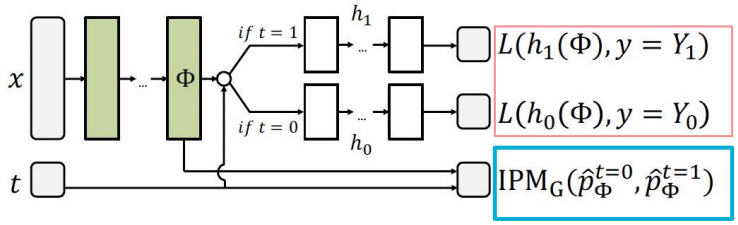


Figure 1. Neural network architecture for ITE estimation. L is a loss function, IPM_G is an integral probability metric. Note that only one of h_0 and h_1 is updated for each sample during training.

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

[FYI]Domain Adversarial Neural Networks (DANN)

Domain Adversarial Neural Networks [Ganin et al., 2016]

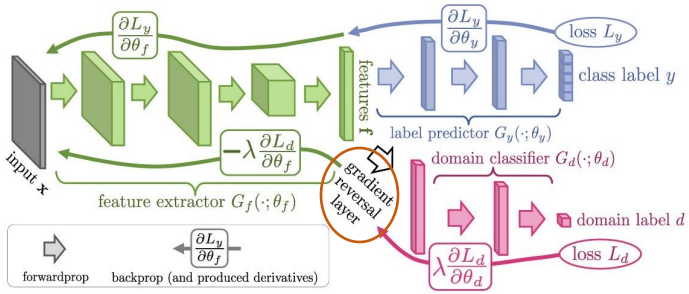


Figure 1: The **proposed architecture** includes a deep *feature extractor* (green) and a deep *label predictor* (blue), which together form a standard feed-forward architecture. Unsupervised domain adaptation is achieved by adding a *domain classifier* (red) connected to the feature extractor via a *gradient reversal layer* that multiplies the gradient by a certain negative constant during the backpropagation-based training. Otherwise, the training proceeds standardly and minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples). Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

DANN's object

$$E(\theta_f,\theta_y,\theta_d)=\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y^i(\theta_f,\theta_y)-\lambda\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_d^i(\theta_f,\theta_d)+\frac{1}{n'}\sum_{i=n+1}^{N}\mathcal{L}_d^i(\theta_f,\theta_d)\right),$$

CFR's object

$$\min_{\substack{h,\Phi\\ \|\Phi\|=1}}\frac{1}{n}\sum_{i=1}^{n}w_i\cdot L\left(h(\Phi(x_i),t_i),y_i\right)+\lambda\cdot\mathfrak{R}(h)$$
$$+\alpha\cdot\mathrm{IPM}_G\left(\{\Phi(x_i)\}_{i:t_i=0},\{\Phi(x_i)\}_{i:t_i=1}\right),$$

[Ganin et al., 2016] Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. Journal of Machine Learning Research 17(1):2096–2030.

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Theoretical Bound (1/3) : Motivation

- Define the ITE error as follows (PEHE)
- However, this is an **unmeasurable metric** (= True ITE is almost impossible to ascertain)
- Existence of a theoretical bound!!

Definition 6. *The expected Precision in Estimation of Heterogeneous Effect (PEHE, Hill (2011)) loss of f is:*

estimated ITE

true ITE

$$\epsilon_{PEHE}(f) = \int_{\mathcal{X}} \left(\hat{\tau}_f(x) - \tau(x)\right)^2 p(x)\, dx, \qquad (1)$$

When $f(x,t) = h(\Phi(x),t)$, we will also use the notation $\epsilon_{PEHE}(h,\Phi) = \epsilon_{PEHE}(f)$.

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Theoretical Bound (2/3) : Definition

- Define **excepted counterfactual loss** to indicate a theoretical bound. This metric also cannot be directly ascertained
- Note that it is different from "expected **factual** treated/control losses".

expected factual treated/control losses

Definition 3. *The expected factual* treated *and* control *losses are:*

$$\epsilon_F^{t=1}(h,\Phi) = \int_{\mathcal{X}} \ell_{h,\Phi}(x,1)\, p^{t=1}(x)\, dx,$$

$$\epsilon_F^{t=0}(h,\Phi) = \int_{\mathcal{X}} \ell_{h,\Phi}(x,0)\, p^{t=0}(x)\, dx.$$

For $u := p(t=1)$, it is immediate to show that $\epsilon_F(h,\Phi) = u\epsilon^{t=1}(h,\Phi) + (1-u)\epsilon^{t=0}(h,\Phi)$.

excepted counterfactual loss

Definition 2. *The expected loss for the unit and treatment pair* (x,t) *is:* $\ell_{h,\Phi}(x,t) = \int_{\mathcal{Y}} L(Y_t, h(\Phi(x),t))p(Y_t|x)dY_t$. *The expected factual and counterfactual losses of* h *and* Φ *are:*

$$\epsilon_F(h,\Phi) = \int_{\mathcal{X}\times\{0,1\}} \ell_{h,\Phi}(x,t)\, p(x,t)\, dxdt,$$

$$\epsilon_{CF}(h,\Phi) = \int_{\mathcal{X}\times\{0,1\}} \ell_{h,\Phi}(x,t)\, p(x,1-t)\, dxdt.$$

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Theoretical Bound (3/3)

By setting an upper bound on CF loss, we were able to show that an upper bound on ITE loss also exists.

Lemma

u := p(t=1), a constant B > 0,
CF loss is given boud by factual loss and IPM (**both observable metrics**)

$$\epsilon_{CF}(h,\Phi) \leq (1-u)\epsilon_F^{t=1}(h,\Phi) + u\epsilon_F^{t=0}(h,\Phi) + B_\Phi \cdot IPM_G\left(p_\Phi^{t=1}, p_\Phi^{t=0}\right),$$

Theorem

The main idea of the proof is showing that ϵ_{PEHE} is upper bounded by the sum of the expected factual loss ϵ_F and expected counterfactual loss ϵ_{CF}. However, we cannot estimate ϵ_{CF}, since we only have samples relevant to ϵ_F. We therefore bound the difference $\epsilon_{CF} - \epsilon_F$ using an IPM.

$$\epsilon_{PEHE}(h,\Phi) \leq 2\left(\epsilon_{CF}(h,\Phi) + \epsilon_F(h,\Phi) - 2\sigma_Y^2\right) \leq \quad (2)$$
$$2\left(\epsilon_F^{t=0}(h,\Phi) + \epsilon_F^{t=1}(h,\Phi) + B_\Phi IPM_G\left(p_\Phi^{t=1}, p_\Phi^{t=0}\right) - 2\sigma_Y^2\right),$$

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

Let’s do it with pytorch!

The following two github repositories were used as references.
The former([cfrnet]) is the official implementation of the original; it is implemented in TensorFlow. The algorithm inside was used as a reference. The latter([SC-CFR]) is implemented in PyTorch.

- [cfrnet] https://github.com/clinicalml/cfrnet
- [SC-CFR] https://github.com/koh-t/SC-CFR

```python
class CFR(Base):
    def __init__(self, in_dim, out_dim, cfg={}):
        super().__init__(cfg)

        self.repnet = MLP(~~~~~)

        if cfg["split_outnet"]:
            self.outnet_treated = MLP(~~~~~)
            self.outnet_control = MLP(~~~~~)
            # Processing here is omitted
        else:
            self.outnet = MLP(~~~~~)
            # Processing here is omitted

        self.optimizer = optim.Adam(
            params=self.params, lr=cfg["lr"], weight_decay=cfg["wd"]
        )
        self.scheduler = StepLR(self.optimizer, step_size=100, gamma=cfg["gamma"])

    def forward(self, x, z):
        with torch.no_grad():

            x_rep = self.repnet(x)

            if self.cfg["split_outnet"]:

                _t_id = np.where((z.cpu().detach().numpy() == 1).all(axis=1))[0]
                _c_id = np.where((z.cpu().detach().numpy() == 0).all(axis=1))[0]

                y_hat_treated = self.outnet_treated(x_rep[_t_id])
                y_hat_control = self.outnet_control(x_rep[_c_id])

                _index = np.argsort(np.concatenate([_t_id, _c_id], 0))
                y_hat = torch.cat([y_hat_treated, y_hat_control])[_index]
            else:
                y_hat = self.outnet(torch.cat((x_rep, z), 1))

        return y_hat
```

repnet

outnet

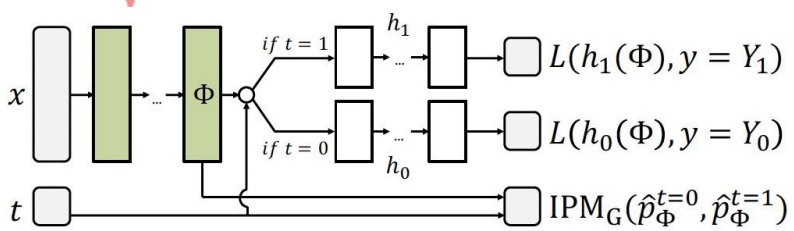


Figure 1. Neural network architecture for ITE estimation. L is a loss function, IPM_G is an integral probability metric. Note that only one of h_0 and h_1 is updated for each sample during training.

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

```python
class Base(nn.Module):
    def __init__(self, cfg):
        super(Base, self).__init__()
        self.cfg = cfg
        self.criterion = nn.MSELoss(reduction='none')

    def fit(self, ~~~~~):
        for epoch in range(self.cfg["epochs"]):
            for (x, y, z) in dataloader:
                # Processing here is omitted
                self.optimizer.zero_grad()

                # Processing here is omitted
                if self.cfg["split_outnet"]:
                    # Processing here is omitted
                    y_hat = torch.cat([y_hat_treated, y_hat_control])[_index]
                else:
                    y_hat = self.outnet(torch.cat((x_rep, z), 1))

                loss = self.criterion(y_hat, y.reshape([-1, 1]))
                # sample weight:Processing here is omitted
                loss =torch.mean((loss * sample_weight))

                if self.cfg["alpha"] > 0.0:
                    if self.cfg["ipm_type"] == "mmd_rbf":
                        # Processing here is omitted
                    elif self.cfg["ipm_type"] == "mmd_lin":
                        ipm = mmd_lin(
                            x_rep[_t_id], x_rep[_c_id],
                            p=len(_t_id) / (len(_t_id) + len(_c_id))
                        )
                    else:
                        # Processing here is omitted
                    loss += ipm * self.cfg["alpha"]
                loss.backward()
                self.optimizer.step()
            # Processing here is omitted
```

$$\min_{\substack{h,\Phi \\ \|\Phi\|=1}} \frac{1}{n}\sum_{i=1}^{n} w_i \cdot L\left(h(\Phi(x_i), t_i), y_i\right) + \lambda \cdot \Re(h)$$

$$+ \alpha \cdot \mathrm{IPM}_G\left(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\right),$$

$$\text{with } w_i = \frac{t_i}{2u} + \frac{1-t_i}{2(1-u)}, \quad \text{where } u = \frac{1}{n}\sum_{i=1}^{n} t_i,$$

and $\Re$ is a model complexity term.

In this experiment, **linear maximum-mean discrepancy** (linear MMD) was employed.

$$\mathrm{MMD} = 2\left\| \frac{1}{m}\sum_{j=1}^{m} \Phi_{\mathbf{W}}(x_{i_j}) - \frac{1}{m'}\sum_{k=m+1}^{m'} \Phi_{\mathbf{W}}(x_{i_k}) \right\|_2$$

Let

$$\mathrm{f}(\mathbf{W}) = \frac{1}{m}\sum_{j=1}^{m} \Phi_{\mathbf{W}}(x_{i_j}) - \frac{1}{m'}\sum_{k=m+1}^{m+m'} \Phi_{\mathbf{W}}(x_{i_k})$$

For other pseudo-distances (e.g. Wasserstein distance), please check the official github
https://github.com/clinicalml/cfrnet

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

Well-known RCT dataset: LaLonde(1986)

- **The National Supported Work Demonstration (NSW)** : The interest of this experiment is whether "vocational training" (counseling and short-term work experience) affects subsequent earnings. In the dataset, the treatment variable, vocational training, is denoted by **treat**, and the outcome variable, income in 1978, is denoted by **re78**.
- Data can be downloaded at the following website: https://users.nber.org/~rdehejia/data/

outcome

treatment 1 or 0

	treat	age	education	black	hispanic	married	nodegree	re74	re75	re78
0	1	37.0	11.0	1.0	0.0	1.0	1.0	0.0	0.0	9930.045898
1	1	22.0	9.0	0.0	1.0	0.0	1.0	0.0	0.0	3595.894043
2	1	30.0	12.0	1.0	0.0	0.0	0.0	0.0	0.0	24909.449219
3	1	27.0	11.0	1.0	0.0	0.0	1.0	0.0	0.0	7506.145996
4	1	33.0	8.0	1.0	0.0	0.0	1.0	0.0	0.0	289.789886

Dehejia, Rajeev and Sadek Wahba. (1999).Causal Effects in Non-Experimental Studies: Re-Evaluating the Evaluation of Training Programs. Journal of the American Statistical Association 94 (448): 1053-1062.
LaLonde, Robert. (1986). Evaluating the Econometric Evaluations of Training Programs. American Economic Review 76:604-620.

[Shalit et al., 2017]'s results

Simulated outcome Real-world outcome

Within-sample

	IHDP		JOBS	
	$\sqrt{\epsilon_{\text{PEHE}}}$	ϵ_{ATE}	R_{POL}	ϵ_{ATT}
OLS/LR-1	$5.8 \pm .3$	$.73 \pm .04$	$.22 \pm .0$	$.01 \pm .00$
OLS/LR-2	$2.4 \pm .1$	$.14 \pm .01$	$.21 \pm .0$	$.01 \pm .01$
BLR	$5.8 \pm .3$	$.72 \pm .04$	$.22 \pm .0$	$.01 \pm .01$
k-NN	$2.1 \pm .1$	$.14 \pm .01$	$.02 \pm .0$	$.21 \pm .01$
TMLE	$5.0 \pm .2$	$.30 \pm .01$	$.22 \pm .0$	$.02 \pm .01$
BART	$2.1 \pm .1$	$.23 \pm .01$	$.23 \pm .0$	$.02 \pm .00$
RAND.FOR.	$4.2 \pm .2$	$.73 \pm .05$	$.23 \pm .0$	$.03 \pm .01$
CAUS.FOR.	$3.8 \pm .2$	$.18 \pm .01$	$.19 \pm .0$	$.03 \pm .01$
BNN	$2.2 \pm .1$	$.37 \pm .03$	$.20 \pm .0$	$.04 \pm .01$
TARNET	$.88 \pm .0$	$.26 \pm .01$	$.17 \pm .0$	$.05 \pm .02$
CFR MMD	$.73 \pm .0$	$.30 \pm .01$	$.18 \pm .0$	$.04 \pm .01$
CFR WASS	$.71 \pm .0$	$.25 \pm .01$	$.17 \pm .0$	$.04 \pm .01$

Out-of-sample

	IHDP		JOBS	
	$\sqrt{\epsilon_{\text{PEHE}}}$	ϵ_{ATE}	R_{POL}	ϵ_{ATT}
OLS/LR-1	$5.8 \pm .3$	$.94 \pm .06$	$.23 \pm .0$	$.08 \pm .04$
OLS/LR-2	$2.5 \pm .1$	$.31 \pm .02$	$.24 \pm .0$	$.08 \pm .03$
BLR	$5.8 \pm .3$	$.93 \pm .05$	$.25 \pm .0$	$.08 \pm .03$
k-NN	$4.1 \pm .2$	$.79 \pm .05$	$.26 \pm .0$	$.13 \pm .05$
BART	$2.3 \pm .1$	$.34 \pm .02$	$.25 \pm .0$	$.08 \pm .03$
RAND.FOR.	$6.6 \pm .3$	$.96 \pm .06$	$.28 \pm .0$	$.09 \pm .04$
CAUS.FOR.	$3.8 \pm .2$	$.40 \pm .03$	$.20 \pm .0$	$.07 \pm .03$
BNN	$2.1 \pm .1$	$.42 \pm .03$	$.24 \pm .0$	$.09 \pm .04$
TARNET	$.95 \pm .0$	$.28 \pm .01$	$.21 \pm .0$	$.11 \pm .04$
CFR MMD	$.78 \pm .0$	$.31 \pm .01$	$.21 \pm .0$	$.08 \pm .03$
CFR WASS	$.76 \pm .0$	$.27 \pm .01$	$.21 \pm .0$	$.09 \pm .03$

$$\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{n}\sum_{i\in[n]}(\tau_i - \hat{\tau}_i)^2}$$

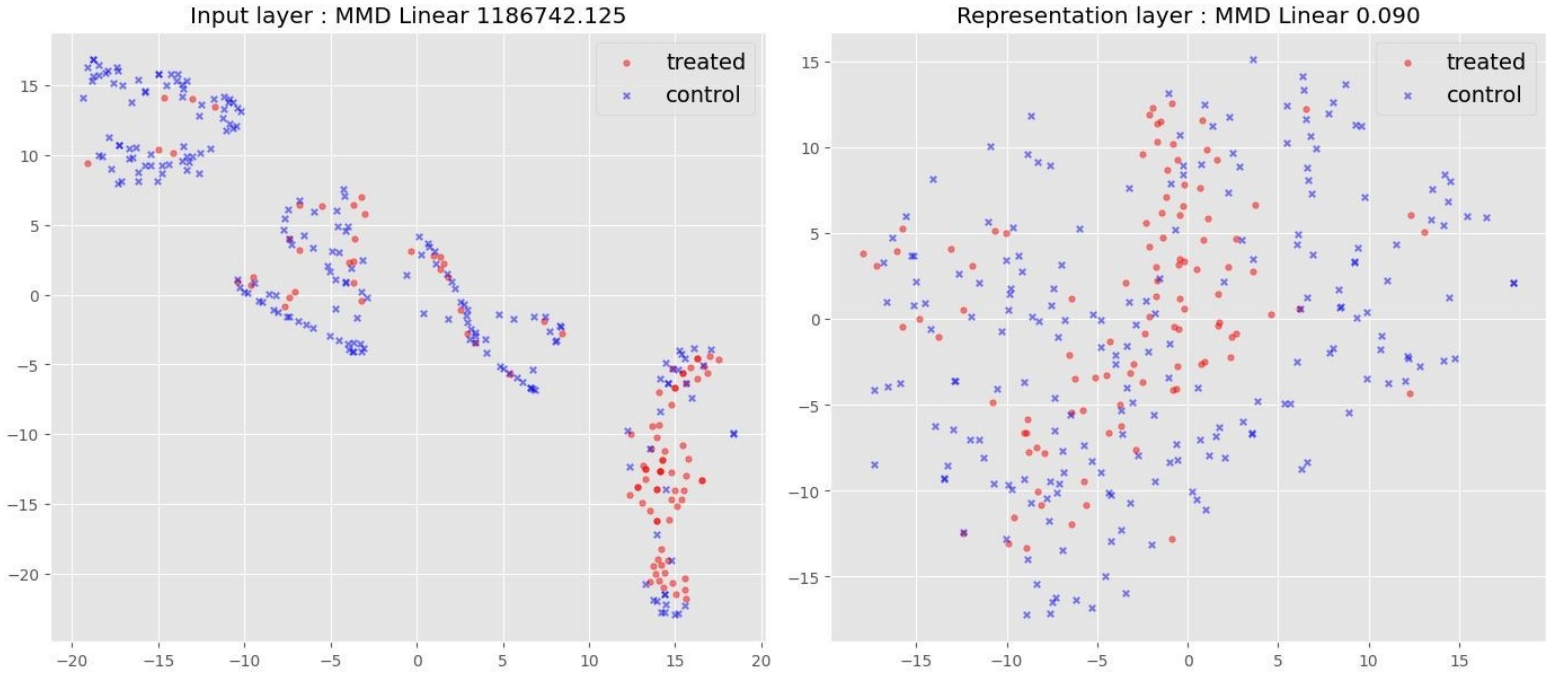
$$\epsilon_{ATE} = \left|\frac{1}{n}\sum_{i\in[n]}\tau_i - \frac{1}{n}\sum_{i\in[n]}\hat{\tau}_i\right|$$

- CFR outperforms baselines in most experiments
- Representation balancing can bring improvement for the results

[Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.

Visualization of Representation Layer

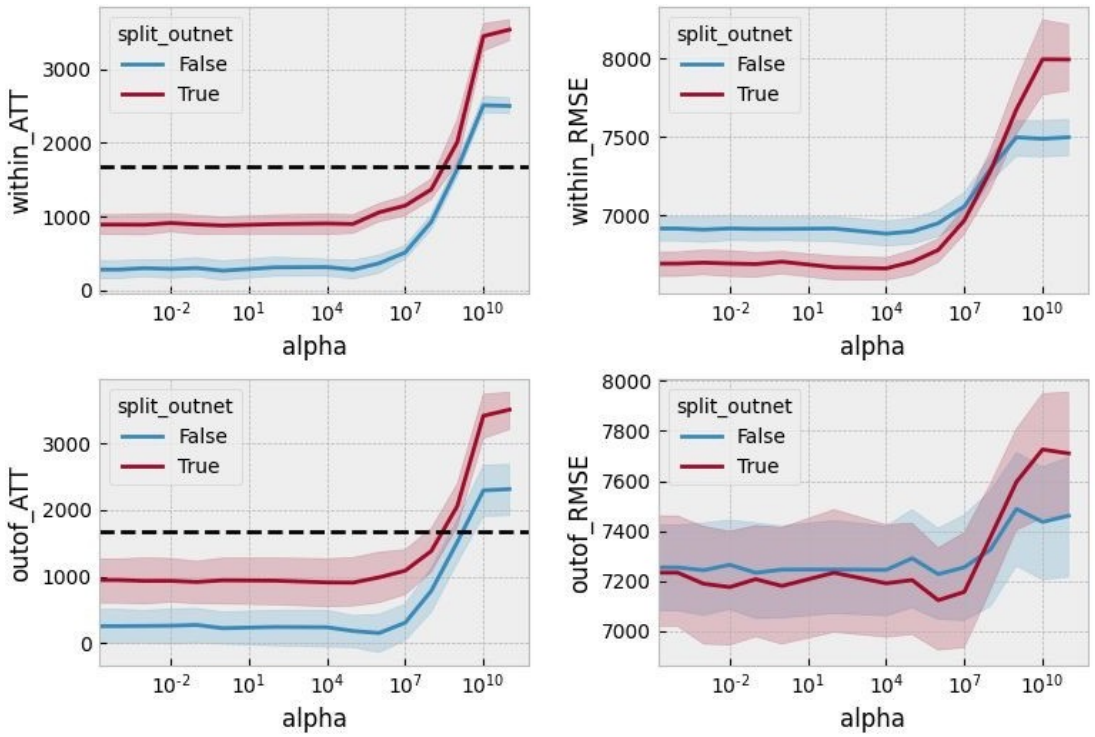
- t-SNE visualizations of the balanced representations of Job dataset
- Original data on the left, representation layer on the right



More Results & Discussions

alpha:

- In this data, there were few alpha benefits. This is consistent with the results of the original paper
- In any case, the difficulty of tuning of α is very high



Lots of interesting applied research!

- As you know, deep Learning is a very expressive model, so the covariates do **not** have to be table data
- For example, the following paper uses CNN to adjust image data as covariates.

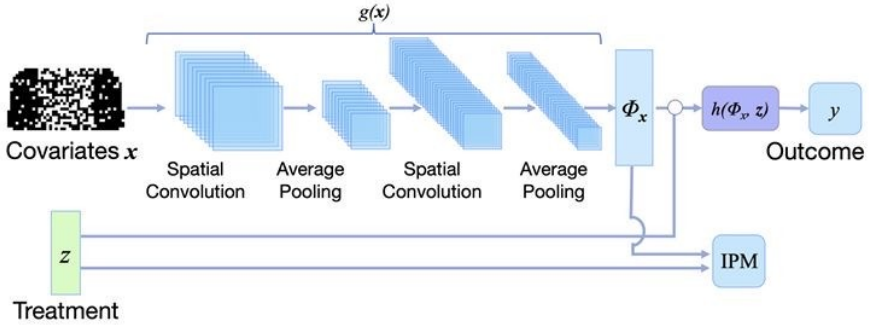


Figure 3: Model architecture of spatial convolutional counterfactual regression (SC-CFR).

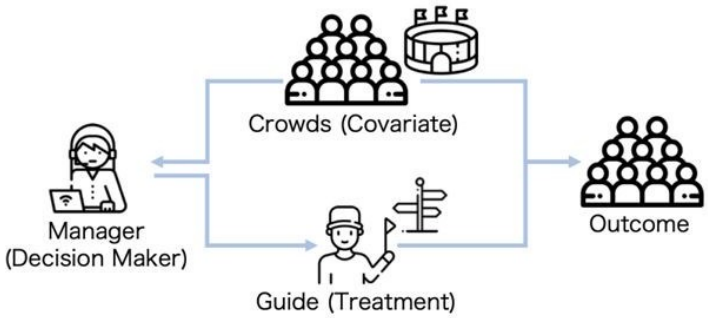


Figure 2: Framework of crowd movement guidance based on causal inferences.

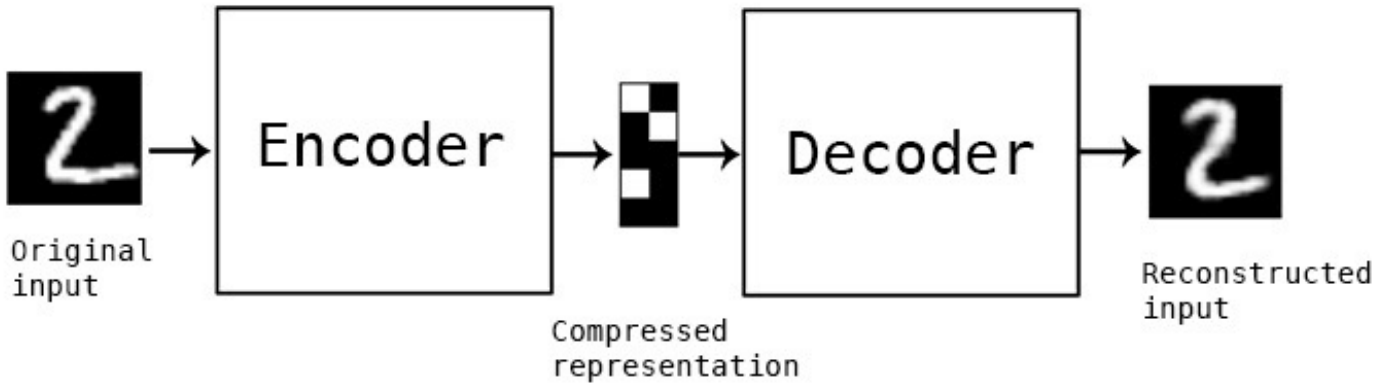
[Takeuchi et al., 2021] Takeuchi, K., Nishida, R., Kashima, H., & Onishi, M. (2021). Grab the Reins of Crowds: Estimating the Effects of Crowd Movement Guidance Using Causal Inference. *arXiv preprint arXiv:2102.03980*.

Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

Autoencoder

- An autoencoder is a type of neural network used to learn efficient codings of unlabeled data (unsupervised learning).
- An autoencoder learns two functions: an encoding function that transforms the input data, and a decoding function that recreates the input data from the encoded representation.
- To make this non-trivial, we need to add a bottleneck layer whose dimension is much smaller than the input.



https://blog.keras.io/building-autoencoders-in-keras.html

Variational Inference

- Suppose we have some distribution q(z). (We'll see later where this comes from.)
- We use Jensen's Inequality to obtain the lower bound.

$$\log p(\mathbf{x}) = \log \int p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z}$$

$$= \log \int q(\mathbf{z})\, \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z}$$

Variational Inference

- Suppose we have some distribution q(z). (We'll see later where this comes from.)
- We use Jensen's Inequality to obtain the lower bound.

$$\begin{aligned}
\log p(\mathbf{x}) &= \log \int p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&= \log \int q(\mathbf{z})\, \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&\geq \int q(\mathbf{z}) \log \left[\frac{p(\mathbf{z})}{q(\mathbf{z})}\, p(\mathbf{x}|\mathbf{z}) \right] \mathrm{d}\mathbf{z} \qquad \text{(Jensen's Inequality)}
\end{aligned}$$

Variational Inference

- Suppose we have some distribution q(z). (We'll see later where this comes from.)
- We use Jensen's Inequality to obtain the lower bound.

$$\begin{aligned}
\log p(\mathbf{x}) &= \log \int p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&= \log \int q(\mathbf{z})\, \frac{p(\mathbf{z})}{q(\mathbf{z})}\, p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&\geq \int q(\mathbf{z}) \log \left[\frac{p(\mathbf{z})}{q(\mathbf{z})}\, p(\mathbf{x}|\mathbf{z}) \right] \mathrm{d}\mathbf{z} \qquad \text{(Jensen's Inequality)} \\
&= \mathbb{E}_q \left[\log \frac{p(\mathbf{z})}{q(\mathbf{z})} \right] + \mathbb{E}_q \left[\log p(\mathbf{x}|\mathbf{z}) \right]
\end{aligned}$$

$-D_{KL}(q(z)||p(z))$, where D_{KL} is Kullback-Leibler (KL) divergence

the expected squared error in reconstructing x from z

Variational Inference

- Hence, we're trying to maximize the variational lower bound (a.k.a. Evidence lower bound, ELBO):

$$\log p(\mathbf{x}) \geq \mathcal{F}(\boldsymbol{\theta}, q) = \mathbb{E}_q \left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{D}_{\mathrm{KL}}(q\|p)$$

- The term "variational" is a historical accident: "variational inference" used to be done using variational calculus, but this isn't how we train VAEs.
- We'd like to choose q to make the bound as tight as possible.
- It's possible to show that the gap is given by:

$$\log p(\mathbf{x}) - \mathcal{F}(\boldsymbol{\theta}, q) = \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x}))$$

- Therefore, we'd like q to be as close as possible to the posterior distribution p(z|x).

Reparameterization Trick

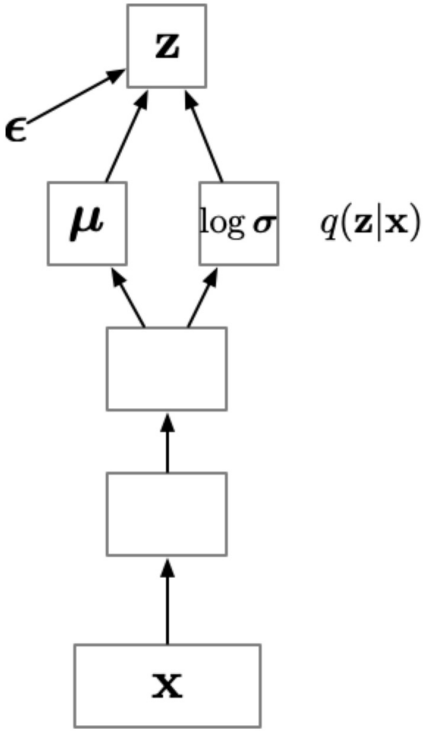
- To fit q, let's assign it a parametric form, in particular a Gaussian distribution: $q(z) = N(z;\ \mu, \Sigma)$, where $\mu = (\mu_1, \ldots, \mu_K)$ and $\Sigma = \ \mathrm{diag}(\sigma_1^2, \ldots, \sigma_K^2)$.
- In general, it's hard to differentiate through an expectation. But for Gaussian q, we can apply the reparameterization trick:

$$z_i = \mu_i + \sigma_i \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, 1)$$

- Hence, $\overline{\mu_i} = \overline{z_i} \qquad \overline{\sigma_i} = \overline{z_i}\epsilon_i$
- This is exactly analogous to how we derived the backpropagation rules for dropout.

Variational Auto Encoder (VAE)

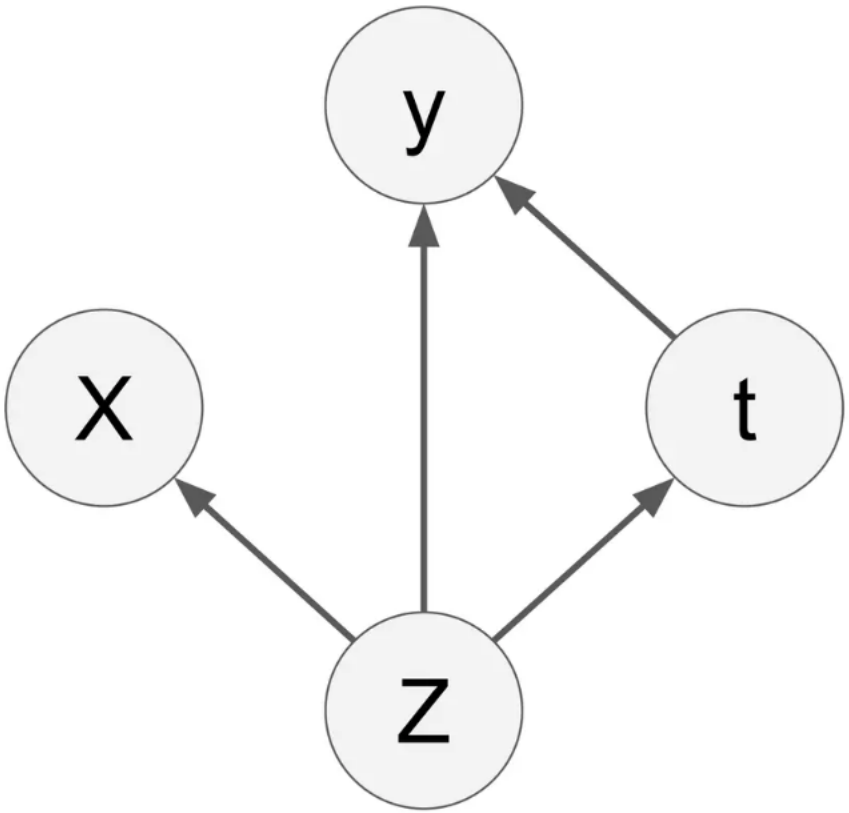
- Idea: amortize the cost of inference by learning an inference network which predicts (μ, Σ) as a function of x.
- The notation q(z|x) emphasizes that q depends on x, even though it's not actually a conditional distribution.
- Combining this with the decoder network, we see the structure closely resembles an ordinary autoencoder. The inference net is like an encoder.
- Hence, this architecture is known as a variational autoencoder (VAE).



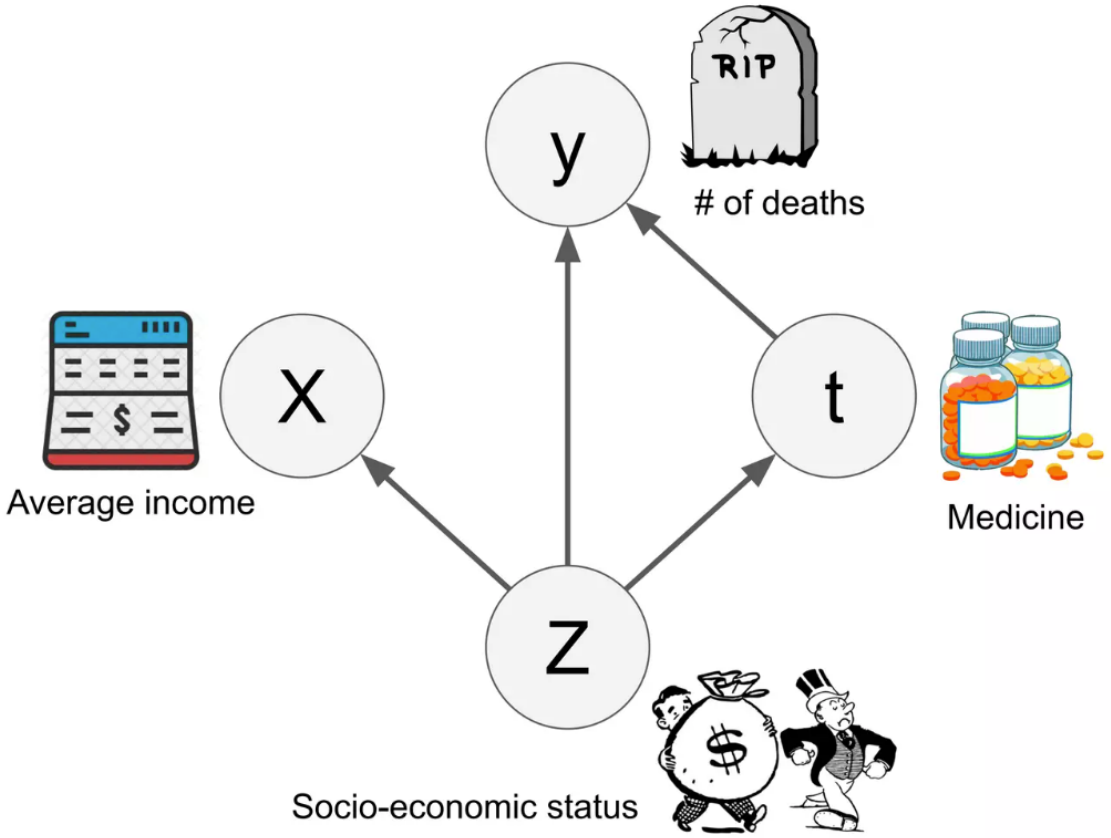
Outline

- Bayesian additive regression trees (BART) for Causal Inference
- Counterfactual Regression (CFR)
  - Model introduction
  - Do it with PyTorch
  - Experiments
- Causal Effect Variational Auto Encoder (CEVAE)
  - Variational autoencoder (VAE)
  - CEVAE

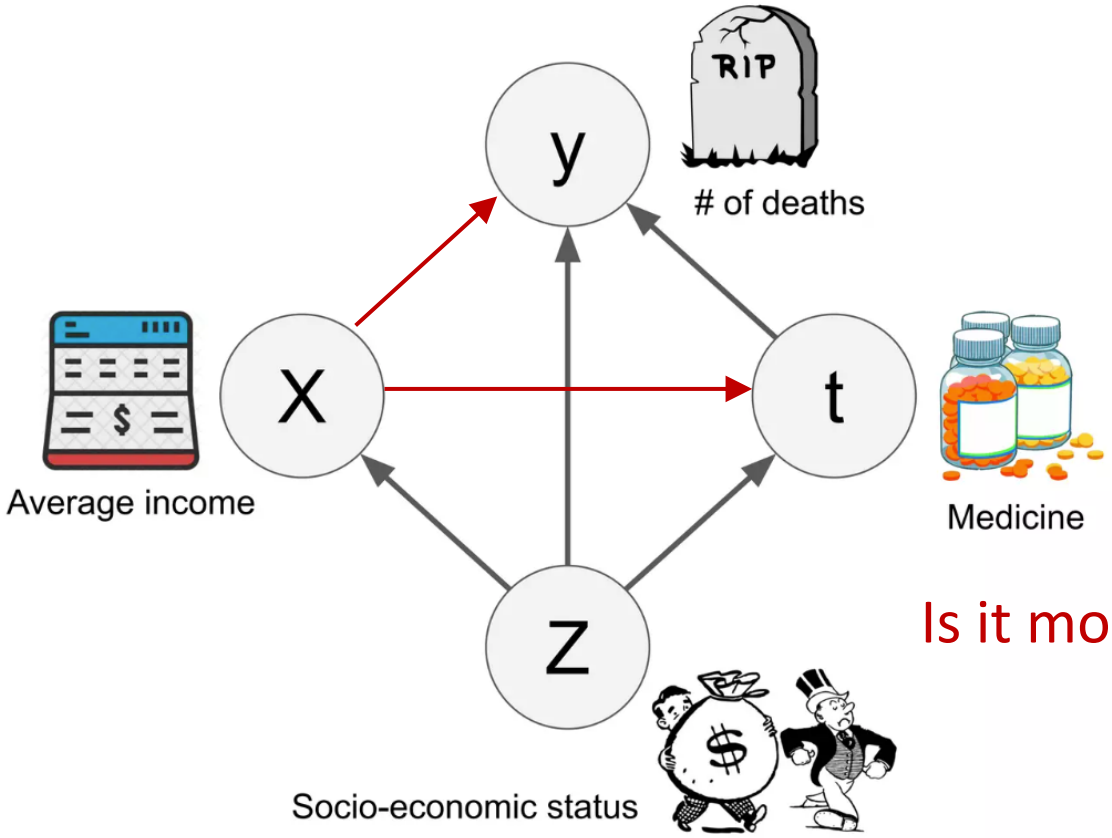
Introduction



Introduction

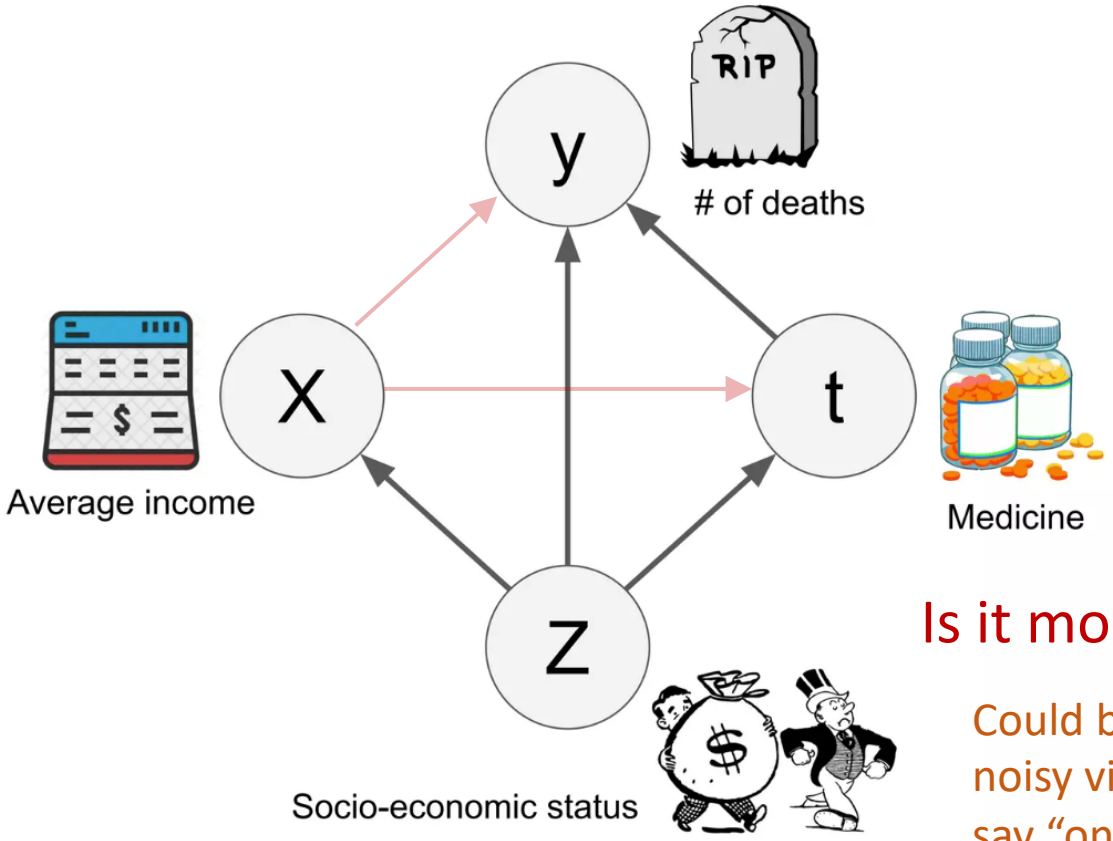


Introduction



Is it more general?

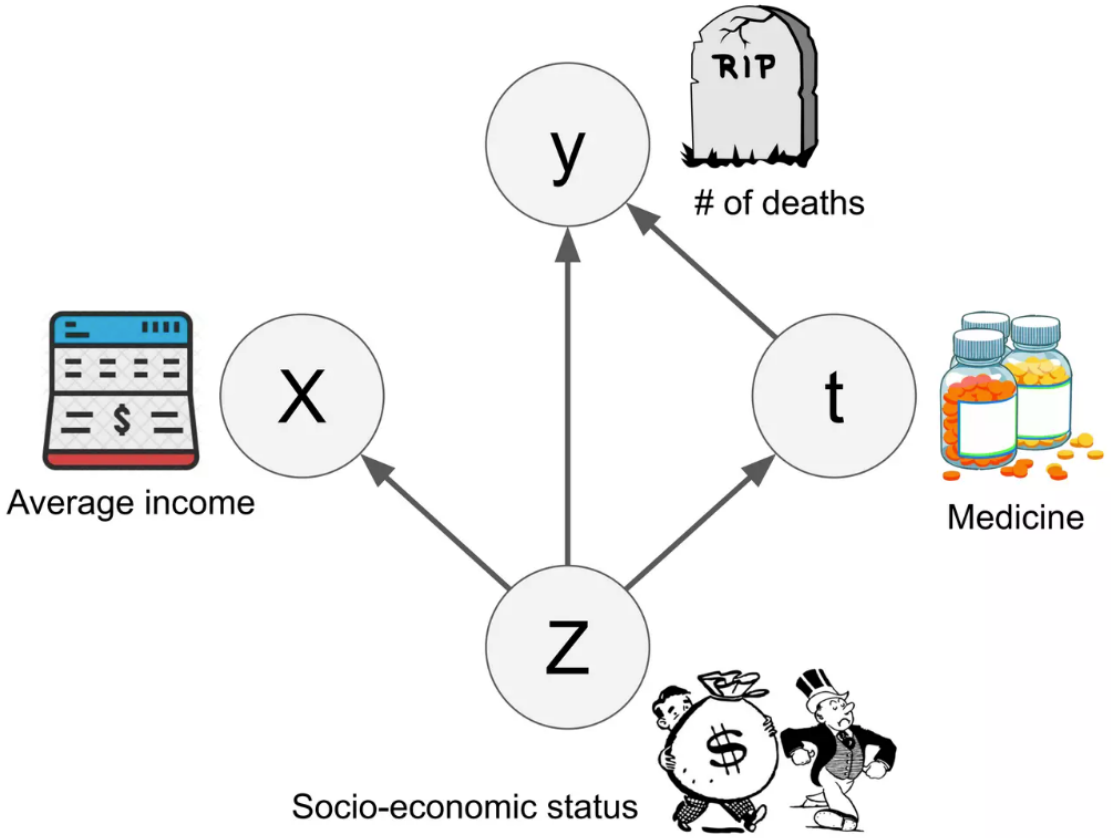
Introduction



Is it more general?

Could be, but X is just a noisy view on Z. We can say “only Z can cause y and t”

Introduction



Identification of Causal Effect

- **Individual Treatment Effect (ITE)**

$$ITE(x) := \mathbb{E}[\mathbf{y}|\mathbf{X} = x, do(\mathbf{t} = 1)] - \mathbb{E}[\mathbf{y}|\mathbf{X} = x, do(\mathbf{t} = 0)]$$

- **Example) "How will the number of deaths(y) vary if we do(t=1) or do not(t=0) treat the poor(Z), whose annual salary(X) is about 10,000 dollars?"**

Do-Calculus

- What if we **set** X to x ?
- **Interventional probability** is generally not the same with **conditional probability**.

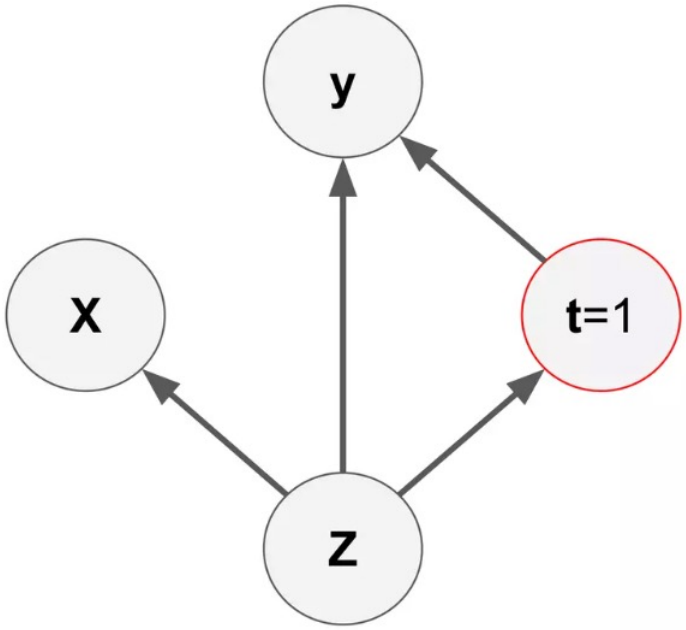
$$\mathbb{P}(Y = y|do(X = x)) \neq \mathbb{P}(Y = y|X = x)$$

- Example)

$$\mathbb{P}(y|do(x)) = \mathbb{P}(y|x)$$

$$\mathbb{P}(y|do(x)) = \mathbb{P}(y)$$

Problem Setup



$$\mathbb{P}(\mathbf{y}|\mathbf{X}, do(\mathbf{t}=1)) = \int_{\mathbf{Z}} \mathbb{P}(\mathbf{y}, \mathbf{Z}|\mathbf{X}, do(\mathbf{t}=1))d\mathbf{Z}$$

Bayes' rule

$$= \int_{\mathbf{Z}} \mathbb{P}(\mathbf{y}|\mathbf{X}, do(\mathbf{t}=1), \mathbf{Z})\mathbb{P}(\mathbf{Z}|\mathbf{X}, do(\mathbf{t}=1))d\mathbf{Z}$$

$$= \int_{\mathbf{Z}} \mathbb{P}(\mathbf{y}|\mathbf{X}, \mathbf{t}=1, \mathbf{Z})\mathbb{P}(\mathbf{Z}|\mathbf{X})d\mathbf{Z}$$

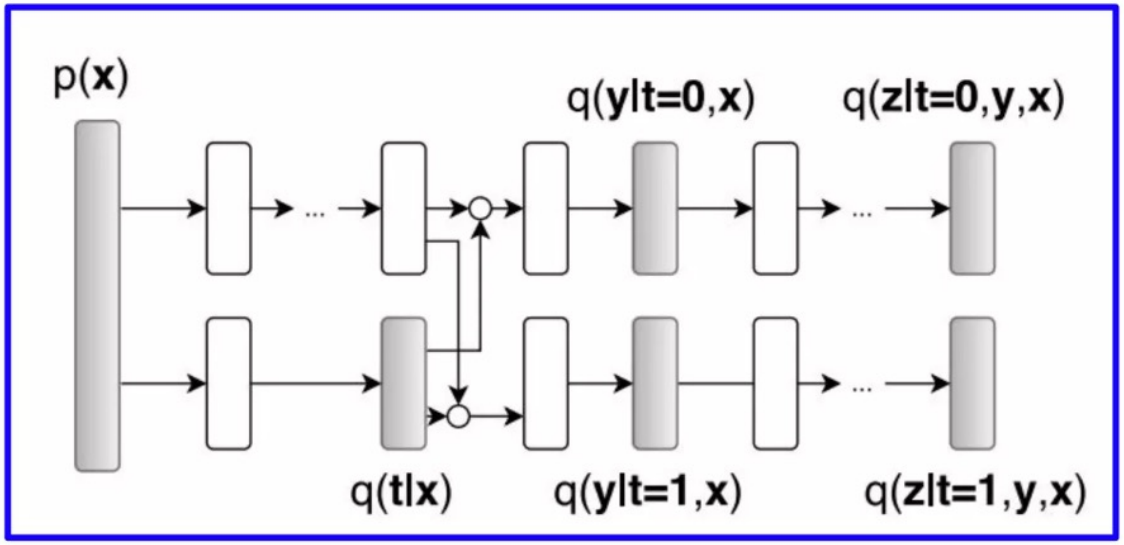
Do-calculus

$$= \mathbb{E}_{p(z|x)}[\mathbb{P}(\mathbf{y}|\mathbf{X}, \mathbf{t}=1, \mathbf{Z})]$$

(The case for t=0 is identical)

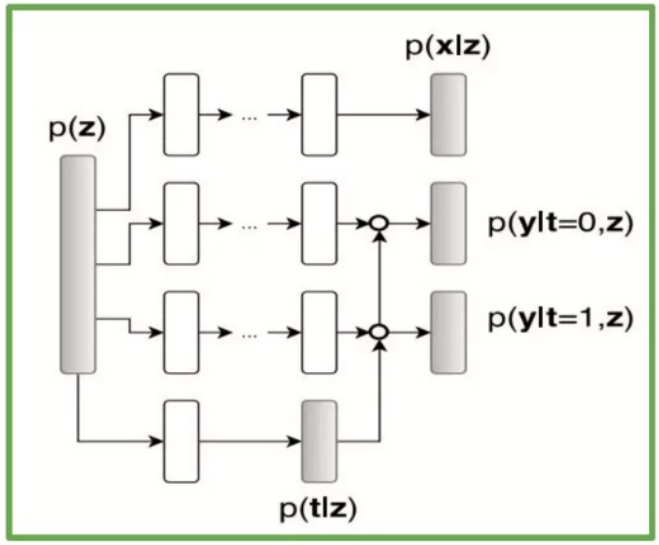
Causal effect variational autoencoder

Parametrize the causal graph as a latent variable model with neural networks



Inference Network

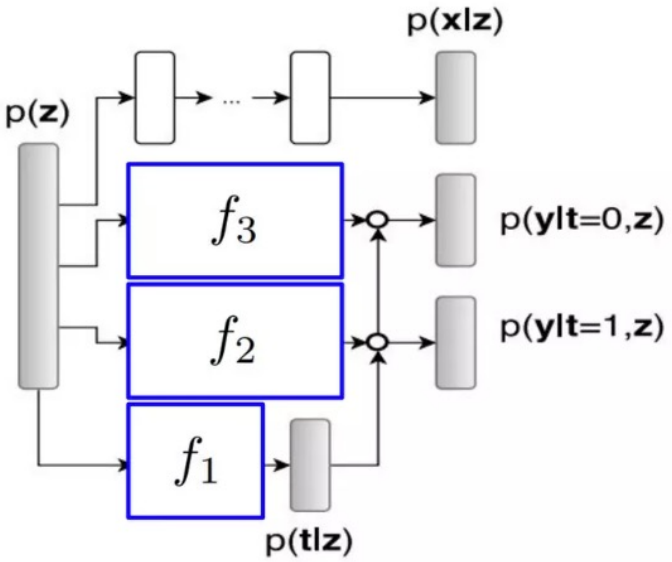
Encoder : model q(z|x,t,y)



Model Network

Decoder : model p(x|z)

Causal effect variational autoencoder

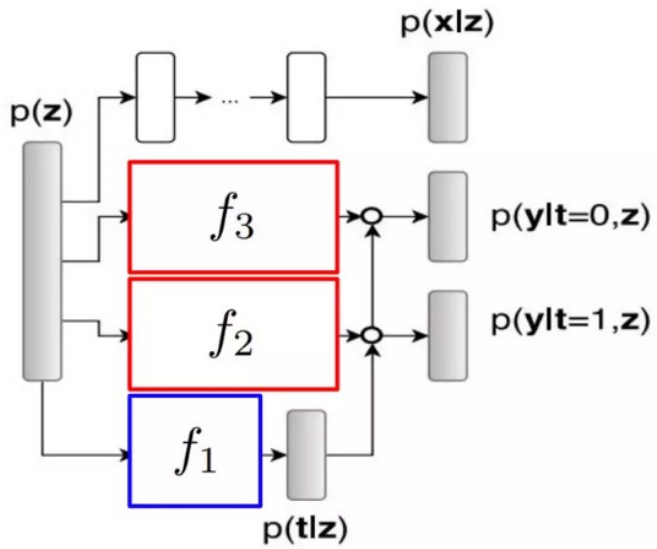


Model Network

- Assume the observations factorize conditioned on the latent variables

$$p(\mathbf{z}_i) = \prod_{j=1}^{D_z} \mathcal{N}(z_{ij}|0,1); \qquad p(\mathbf{x}_i|\mathbf{z}_i) = \prod_{j=1}^{D_x} p(x_{ij}|\mathbf{z}_i);$$

Causal effect variational autoencoder



Model Network

- First, compute the distribution p(t|z) and sample t

$$p(t_i|\mathbf{z}_i) = \text{Bern}(\sigma(f_1(\mathbf{z}_i)))$$

- Next, compute p(y|t,z) and sample y
  - For a continuous outcome: Gaussian distribution

$$p(y_i|t_i, \mathbf{z}_i) = \mathcal{N}(\mu = \hat{\mu}_i, \sigma^2 = \hat{v}) \qquad \hat{\mu}_i = t_i f_2(\mathbf{z}_i) + (1 - t_i) f_3(\mathbf{z}_i)$$

  - For a discrete outcome: Bernoulli distribution.

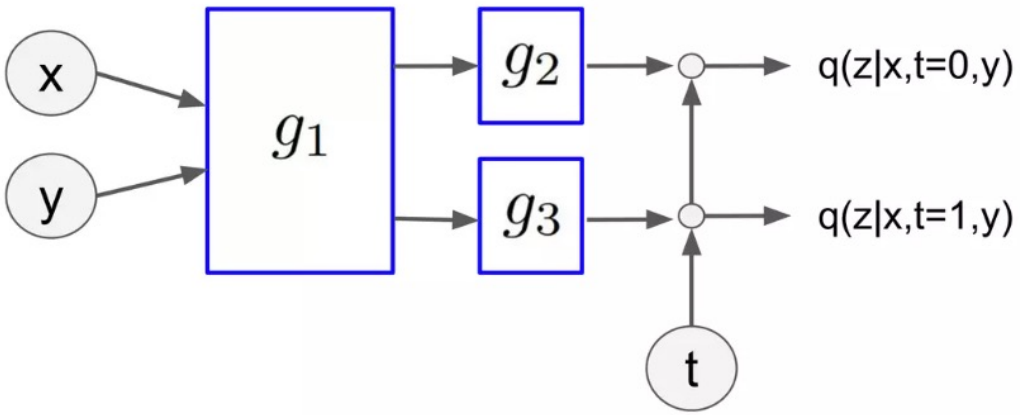
$$p(y_i|t_i, \mathbf{z}_i) = \text{Bern}(\pi = \hat{\pi}_i) \qquad \hat{\pi}_i = \sigma(t_i f_2(\mathbf{z}_i) + (1 - t_i) f_3(\mathbf{z}_i))$$

- Then compute p(x|z) for reconstruction

Causal effect variational autoencoder

Neural networks outputs the parameters of a posterior approximation over the latent variables z, e.g. a Gaussian

$$q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i) = \prod_{j=1}^{D_z} \mathcal{N}(\mu_j = \bar{\mu}_{ij}, \sigma_j^2 = \bar{\sigma}_{ij}^2)$$



Inference Network

For the mean parameters

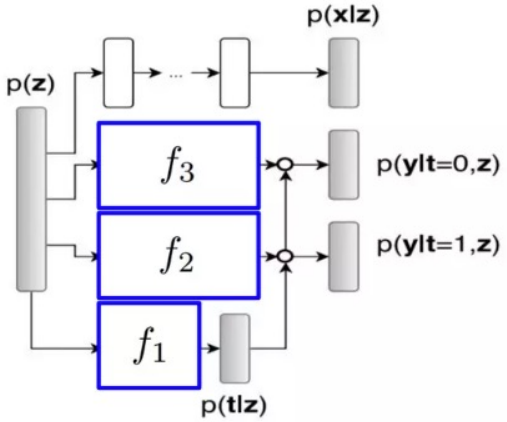
$$\bar{\boldsymbol{\mu}}_i = t_i \boldsymbol{\mu}_{t=0,i} + (1 - t_i)\boldsymbol{\mu}_{t=1,i}$$

For the variance parameters

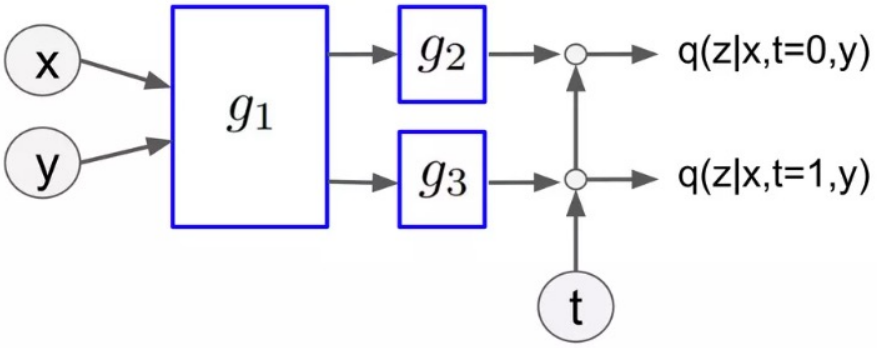
$$\bar{\boldsymbol{\sigma}}_i^2 = t_i \boldsymbol{\sigma}_{t=0,i}^2 + (1 - t_i)\boldsymbol{\sigma}_{t=1,i}^2$$

$$\boldsymbol{\mu}_{t=0,i}, \boldsymbol{\sigma}_{t=0,i}^2 = g_2 \circ g_1(\mathbf{x}_i, y_i)$$
$$\boldsymbol{\mu}_{t=1,i}, \boldsymbol{\sigma}_{t=1,i}^2 = g_3 \circ g_1(\mathbf{x}_i, y_i)$$

Causal effect variational autoencoder



Model Network



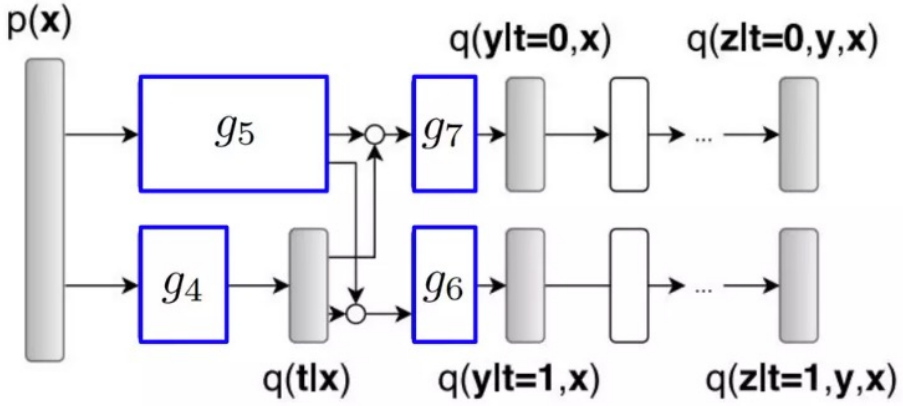
Inference Network

The objective from original VAE - ELBO

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i)}[\log p(\mathbf{x}_i, t_i|\mathbf{z}_i) + \log p(y_i|t_i,\mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i)].$$

Causal effect variational autoencoder

- We need two auxiliary distributions that predict t, y for new samples.



Inference Network

$$q(t_i|\mathbf{x}_i) = \text{Bern}(\pi = \sigma(g_4(\mathbf{x}_i)))$$

- For a continuous outcome: Gaussian distribution

$$q(y_i|\mathbf{x}_i, t_i) = \mathcal{N}(\mu = \bar{\mu}_i, \sigma^2 = \bar{v})$$
$$\bar{\mu}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i))$$

- For a discrete outcome: Bernoulli distribution.

$$q(y_i|\mathbf{x}_i, t_i) = \text{Bern}(\pi = \bar{\pi}_i)$$
$$\bar{\pi}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i))$$

Causal effect variational autoencoder

- ELBO term

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i)}[\log p(\mathbf{x}_i, t_i|\mathbf{z}_i) + \log p(y_i|t_i, \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i)]$$

- The Objective of the Causal Effect Variational Autoencoder (CEVAE)

$$\mathcal{F}_{\text{CEVAE}} = \mathcal{L} + \sum_{i=1}^{N} \left(\log q(t_i = t_i^*|\mathbf{x}_i^*) + \log q(y_i = y_i^*|\mathbf{x}_i^*, t_i^*) \right)$$

Q: Why this two extra terms are needed?

Causal effect variational autoencoder

- The Objective of the Causal Effect Variational Autoencoder (CEVAE)

$$\mathcal{F}_{\text{CEVAE}} = \mathcal{L} + \sum_{i=1}^{N} \left(\log q(t_i = t_i^* | \mathbf{x}_i^*) + \log q(y_i = y_i^* | \mathbf{x}_i^*, t_i^*) \right)$$

Q: Why this two extra terms are needed?

- For unseen data x, we require to know the treatment assignment t and outcome y before inferring the distribution over z.
- So we need two auxiliary distributions that predict t, y for new samples, and two extra terms are for estimating the parameters of these distributions.

Experiment - Dataset

Three main experiment in the present paper

1. Two existing benchmark datasets
    - IHDP (Infant Health and Development Program), Jobs
2. a synthetic toy dataset
3. Introduce a new benchmark
    - based on twin births and deaths in the USA

Experiment - Implementation

- neural network architecture
  - 3 hidden layers NN with ELU nonlinearities
    - approximate posterior over the latent variables $q(Z|X,t,y)$
    - generative model $p(X|Z)$
    - outcome models $p(y|t, Z)$, $q(y|t, X)$.
  - a single hidden layer NN with ELU nonlinearities
    - Treatment models $p(t|Z)$, $q(t|X)$
- latent variable
  - 20 dimensional latent variable z
  - weight decay term for all of the parameters

Experiment - Baseline models

- LR1 = logistic regression
- LR2 = two separate logistic regressions fit
  - to treated (t=1)
  - to control (t=0)
- TARnet
  - FFNN for causal inference

Experiment 1 - Benchmark datasets

- IHDP (Infant Health and Development Program)
- effect of home visits by specialists on future cognitive test scores

- Metrics
  - PEHE (Precision in Estimation of Heterogeneous Effect)

$$\text{PEHE} = \frac{1}{N}\sum_{i=1}^{N}((y_{i1} - y_{i0}) - (\hat{y}_{i1} - \hat{y}_{i0}))^2,$$

  - absolute error on ATE (Average Treatment Effect)

$$ATE := \mathbb{E}[ITE(x)] \qquad ITE(x) := \mathbb{E}\left[\mathbf{y}|\mathbf{X} = x, do(\mathbf{t} = 1)\right] - \mathbb{E}\left[\mathbf{y}|\mathbf{X} = x, do(\mathbf{t} = 0)\right]$$

Experiment 1 - Benchmark datasets

Table 1: Within-sample and out-of-sample mean and standard errors for the metrics for the various models at the IHDP dataset.

Method	$\sqrt{\epsilon_{\text{PEHE}}^{\text{within-s.}}}$	$\epsilon_{\text{ATE}}^{\text{within-s.}}$	$\sqrt{\epsilon_{\text{PEHE}}^{\text{out-of-s.}}}$	$\epsilon_{\text{ATE}}^{\text{out-of-s.}}$
OLS-1	5.8±.3	.73±.04	5.8±.3	.94±.06
OLS-2	2.4±.1	.14±.01	2.5±.1	.31±.02
BLR	5.8±.3	.72±.04	5.8±.3	.93±.05
k-NN	2.1±.1	.14±.01	4.1±.2	.79±.05
TMLE	5.0±.2	.30±.01	-	-
BART	2.1±.1	.23±.01	2.3±.1	.34±.02
RF	4.2±.2	.73±.05	6.6±.3	.96±.06
CF	3.8±.2	.18±.01	3.8±.2	.40±.03
BNN	2.2±.1	.37±.03	2.1±.1	.42±.03
CFRW	.71±.0	.25±.01	.76±.0	.27±.01
CEVAE	2.7±.1	.34±.01	2.6±.1	.46±.02

Experiment 1 - Benchmark datasets

- the effect of job training (treatment) on employment after training (outcome)

- Metrics
  - absolute error on **A**verage **T**reatment effect on the **T**reated (ATT)

$$\mathbb{E}\left[ITE(\mathbf{X})|\mathbf{t}=1\right]$$

  - Policy risk
    - acts as a proxy to the individual treatment effect.

Experiment 1 - Benchmark datasets

Table 2: Within-sample and out-of-sample policy risk and error on the average treatment effect on the treated (ATT) for the various models on the Jobs dataset.

Method	$R_{pol}^{\text{within-s.}}$	$\epsilon_{\text{ATT}}^{\text{within-s.}}$	$R_{pol}^{\text{out-of-s.}}$	$\epsilon_{\text{ATT}}^{\text{out-of-s.}}$
LR-1	.22±.0	.01±.00	.23±.0	.08±.04
LR-2	.21±.0	.01±.01	.24±.0	.08±.03
BLR	.22±.0	.01±.01	.25±.0	.08±.03
k-NN	.02±.0	.21±.01	.26±.0	.13±.05
TMLE	.22±.0	.02±.01	-	-
BART	.23±.0	.02±.00	.25±.0	.08±.03
RF	.23±.0	.03±.01	.28±.0	.09±.04
CF	.19±.0	.03±.01	.20±.0	.07±.03
BNN	.20±.0	.04±.01	.24±.0	.09±.04
CFRW	.17±.0	.04±.01	.21±.0	.09±.03
CEVAE	.15±.0	.02±.01	.26±.0	.03±.01

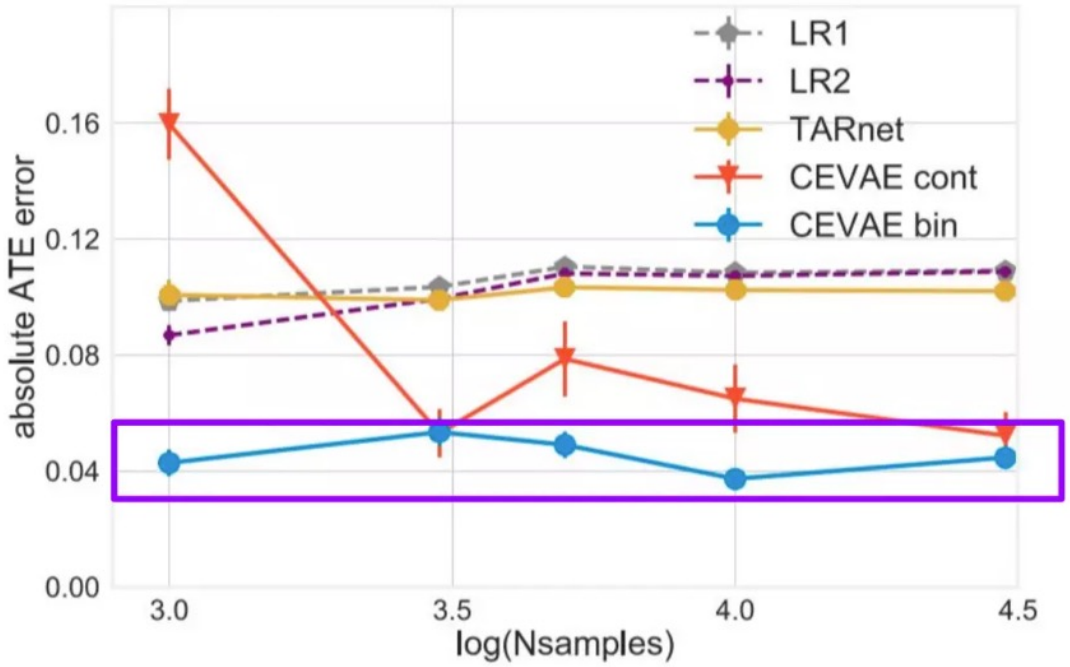
Experiment 2 - Synthetic experiment on toy data

- the marginal distribution of X is a mixture of Gaussians
- the hidden variable Z is determining the mixture component

$$\mathbf{z}_i \sim \text{Bern}\left(0.5\right); \qquad \mathbf{x}_i|\mathbf{z}_i \sim \mathcal{N}\left(\mathbf{z}_i, \sigma_{z_1}^2 \mathbf{z}_i + \sigma_{z_0}^2 (1-\mathbf{z}_i)\right)$$
$$\mathbf{t}_i|\mathbf{z}_i \sim \text{Bern}\left(0.75\mathbf{z}_i + 0.25(1-\mathbf{z}_i)\right); \qquad \mathbf{y}_i|\mathbf{t}_i, \mathbf{z}_i \sim \text{Bern}\left(\text{Sigmoid}\left(3(\mathbf{z}_i + 2(2\mathbf{t}_i - 1))\right)\right)$$

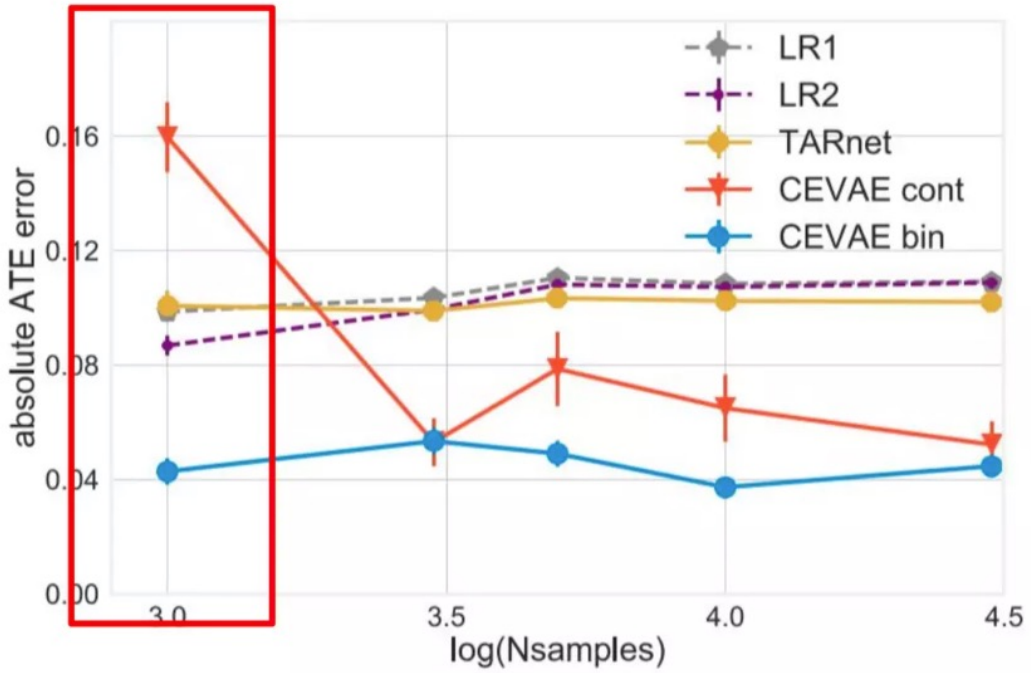
- latent variable z => **binary variable**
- **How models can imitate the true latent variable well**

Experiment 2 - Synthetic experiment on toy data



- CEVAE bin: 5-dim binary latent z
  - latent model is correctly specified
  - better with all sample size

Experiment 2 - Synthetic experiment on toy data



- CEVAE cont:

  5-dim continuous latent z
  - latent model is not correctly specified

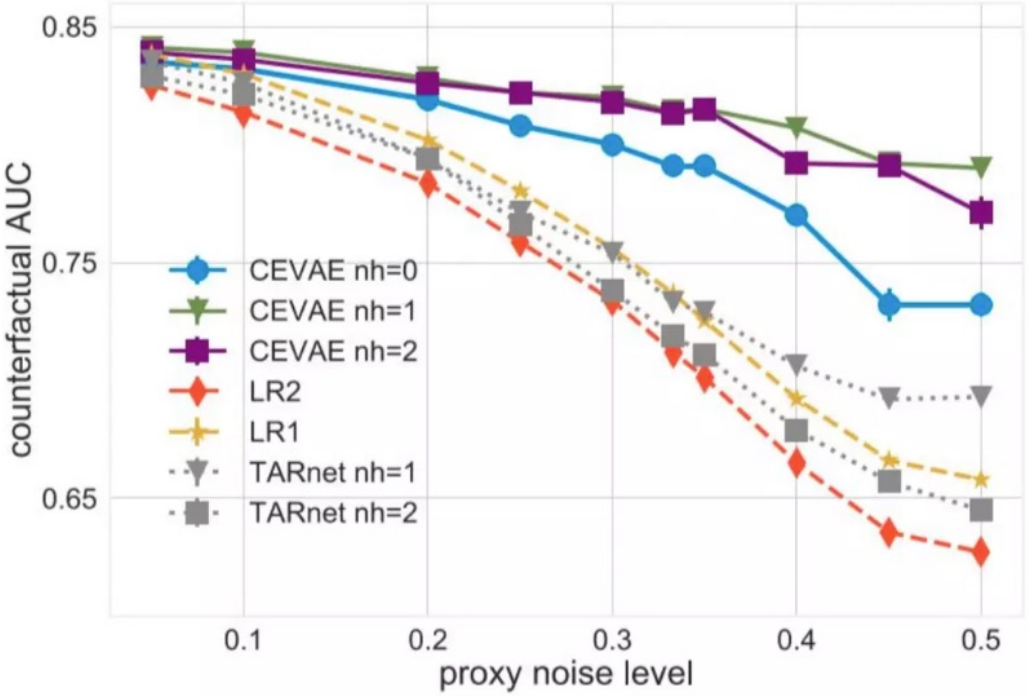
- **We require more samples** for the latent space to imitate more closely the true latent variable

Experiment 3 - Binary treatment outcome on Twins

- Introduce a new benchmark dataset about twin births in the USA
- t = 1
  - being born the heavier than the other.
- Y (outcome)
  - mortality of each of the twins in their first year of life
- Z (latent variable)
  - the number of gestation weeks (20 weeks, 20-27, 27-34, …) **10 categories**
- X (proxy variables)
  - noisy view of Z
  - encoded vector (Z) flipped with a probability of 0.05-0.5
  - 0.5 flip means no direct information from Z

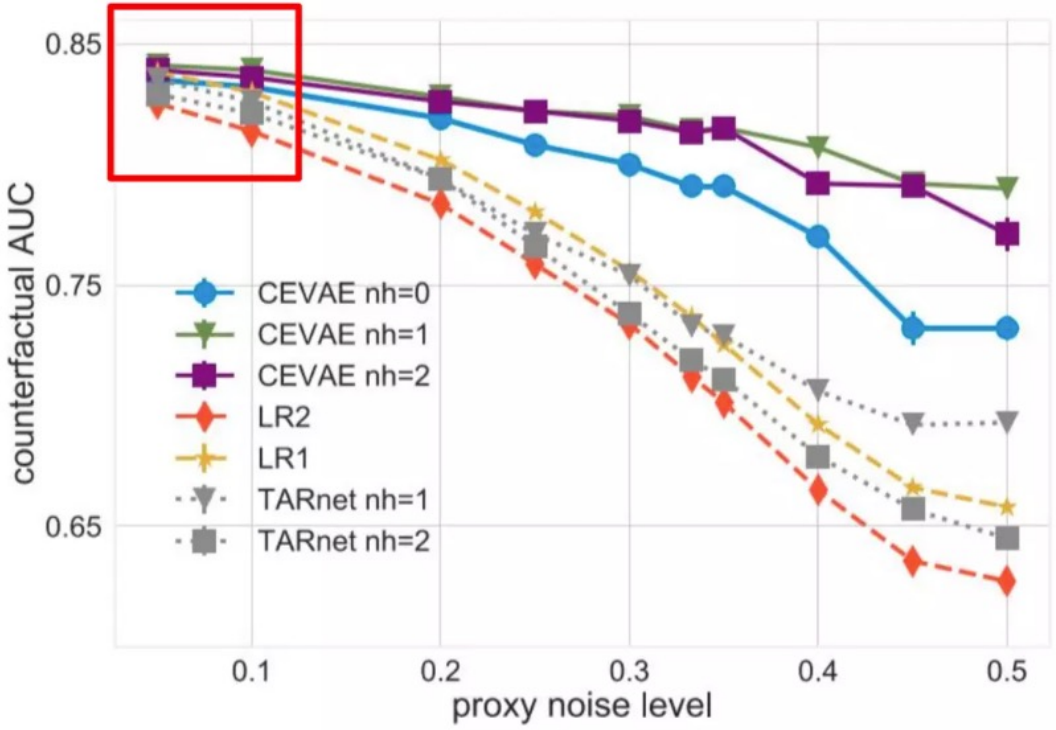
Experiment 3 - Binary treatment outcome on Twins

- inferring the mortality of the unobserved twin (counterfactual)



Experiment 3 - Binary treatment outcome on Twins

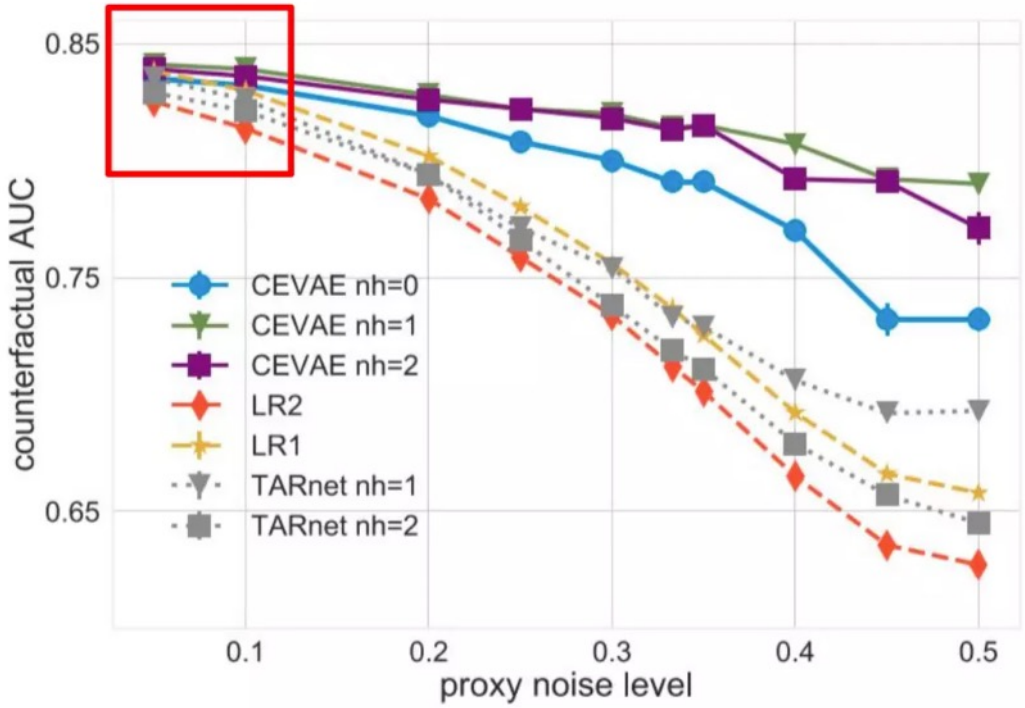
- inferring the mortality of the unobserved twin (counterfactual)



Q: Why does all methods perform similarly when the proxy noise is small?

Experiment 3 - Binary treatment outcome on Twins

- inferring the mortality of the unobserved twin (counterfactual)



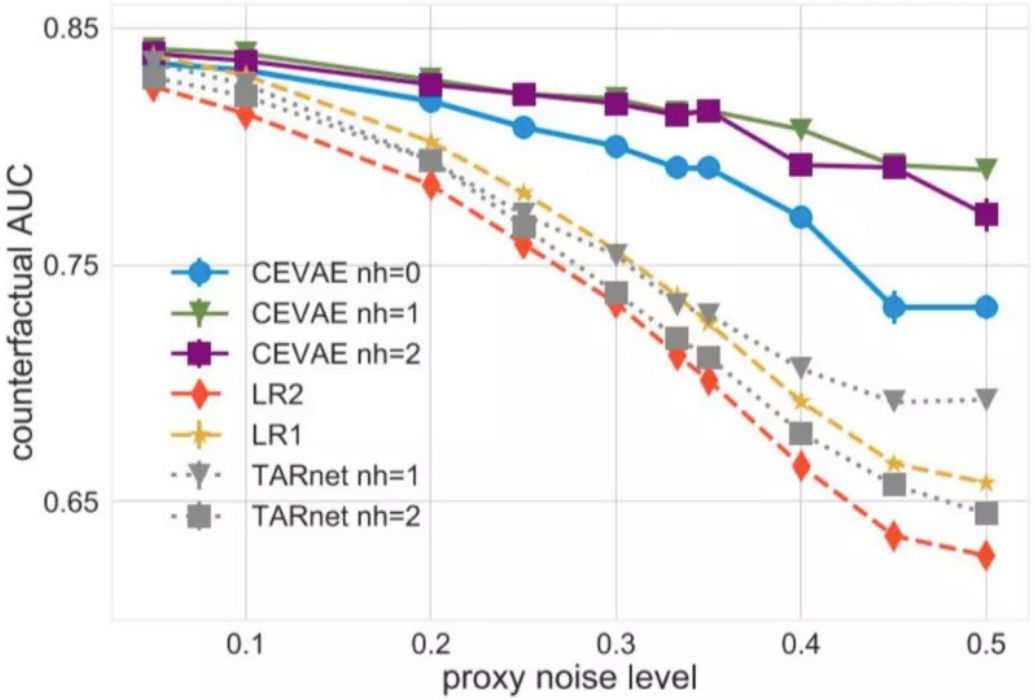
Q: Why does all methods perform similarly when the proxy noise is small?

- X ≈ Z
- Only CEVAE uses Z
- The others use only X

- Z (the gestation length feature) is very informative

Experiment 3 - Binary treatment outcome on Twins

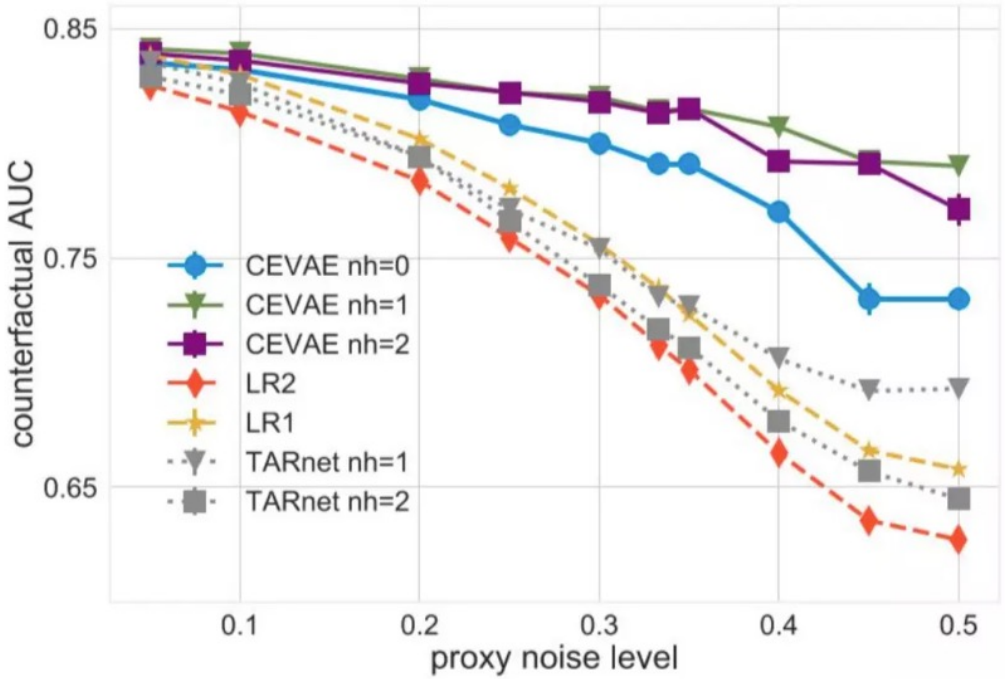
- inferring the mortality of the unobserved twin (counterfactual)



- nh = number of hidden layers
- in CEVAE cases, larger nh, better AUC

Experiment 3 - Binary treatment outcome on Twins

- inferring the mortality of the unobserved twin (counterfactual)

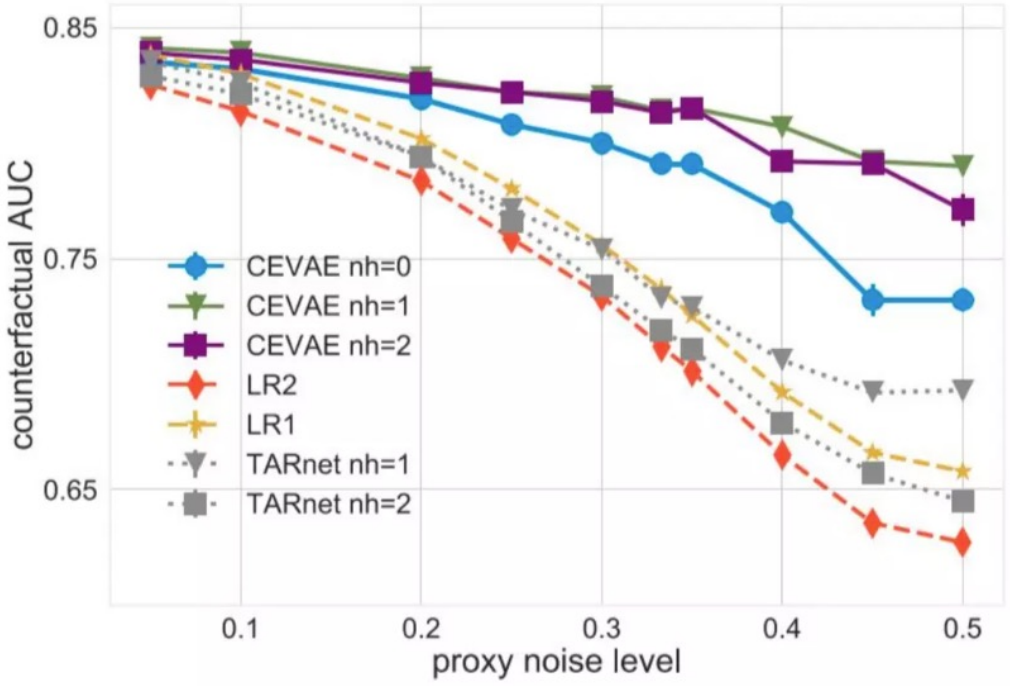


- nh = number of hidden layers
- in CEVAE cases, larger nh, better AUC

Q: “TARnet (nh=0) == LR2“, but Why are [CEVAE nh=0] and [LR2] performing differently when more proxy noisy level?

Experiment 3 - Binary treatment outcome on Twins

- inferring the mortality of the unobserved twin (counterfactual)



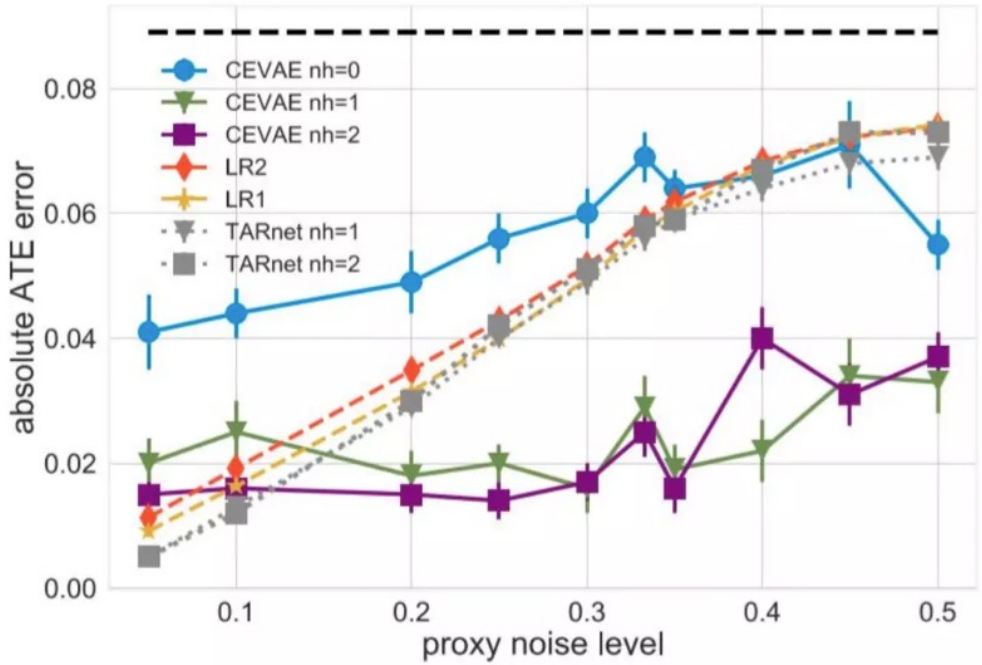
- nh = number of hidden layers
- in CEVAE cases, larger nh, better AUC

Q: “TARnet (nh=0) == LR2“, but Why are [CEVAE nh=0] and [LR2] performing differently when more proxy noisy level?

- LR2 rely directly on the **noisy proxies** instead of **the inferred latent state.**

Experiment 3 - Binary treatment outcome on Twins

- inferring the average treatment effect



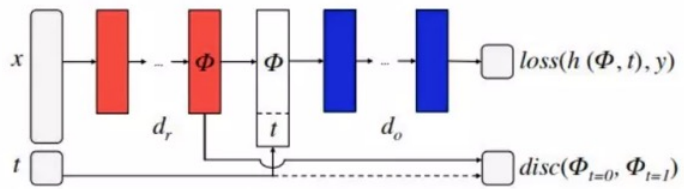
number of hidden layers

- CEVAE nh=0: not so good
- CEVAE is robust with increasing proxy noise

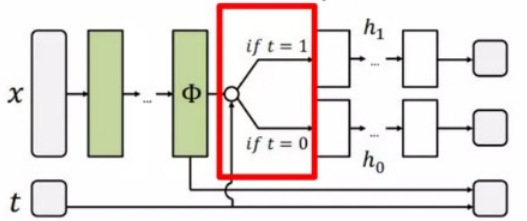
Balancing Neural Network v.s. CEVAE

- Model choice
  - Discriminative model vs. Generative model
  - CEVAE learns latent variable z (i.e. unobserved confounder), which BNN does not learn

- Architecture
  - CEVAE split outputs for each treatment group in t after a shared representation



BNN



CEVAE (~ TARnet architecture)

It can learn the difference of t more explicitly

References

(Papers and links)

- [Shalit et al., 2017] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017
- [Johansson et al., 2016] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." International conference on machine learning. PMLR, 2016.
- [Takeuchi et al., 2021] Takeuchi, Koh, et al. "Grab the Reins of Crowds: Estimating the Effects of Crowd Movement Guidance Using Causal Inference." *arXiv preprint arXiv:2102.03980*, 2021.
- [Ganin et al., 2016] Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. "Domain-adversarial training of neural networks." Journal of Machine Learning Research 17(1):2096–2030, 2016.
- https://www.slideshare.net/ssuserd37bda/causal-effect-inference-with-deep-latentvariable-models
- https://speakerdeck.com/masa_asa/quick-introduction-to-counterfactual-regression-cfr
- https://www.cs.toronto.edu/~rgrosse/courses/csc421_2019/slides/lec17.pdf

(Python code)

- [cfrnet] https://github.com/clinicalml/cfrnet
- [SC-CFR] https://github.com/koh-t/SC-CFR

Reading Materials

- Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." International Conference on Machine Learning. PMLR, 2017.
- Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." International conference on machine learning. PMLR, 2016.
- Louizos C, Shalit U, Mooij J M, et al. Causal effect inference with deep latent-variable models[J]. Advances in neural information processing systems, 2017, 30.

Thank you!