Learning to act in noisy contexts using deep proxy learning

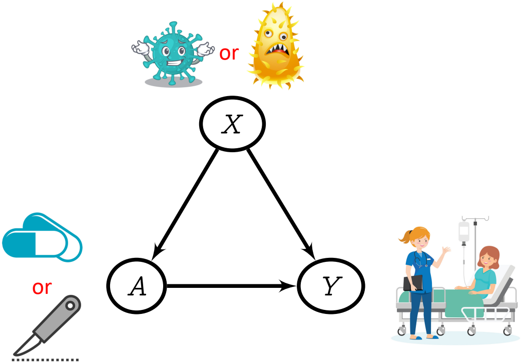
Arthur Gretton

Gatsby Computational Neuroscience Unit
Google Deepmind

FIMI, 2024

Observation vs intervention

Conditioning from observation: $\mathbb{E}[Y|A=a] = \sum_x \mathbb{E}[Y|a,x]p(x|a)$

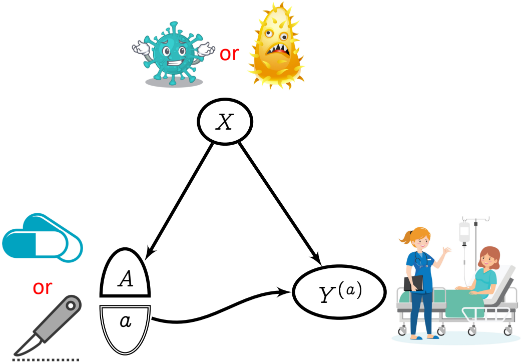


From our *observations* of historical hospital data:

- $P(Y = \text{cured}|A = \text{pills}) = 0.85$
- $P(Y = \text{cured}|A = \text{surgery}) = 0.72$

Observation vs intervention

Average causal effect (intervention): $\mathbb{E}[Y^{(a)}] = \sum_x \mathbb{E}[Y \mid a, x] p(x)$

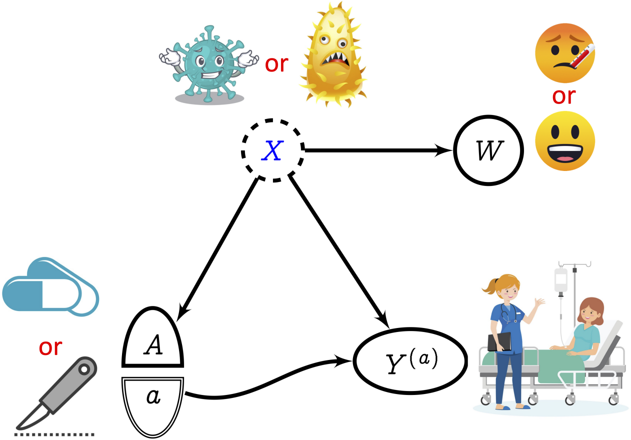


From our *intervention* (making all patients take a treatment):

- $P(Y^{(\text{pills})} = \text{cured}) = 0.64$
- $P(Y^{(\text{surgery})} = \text{cured}) = 0.75$

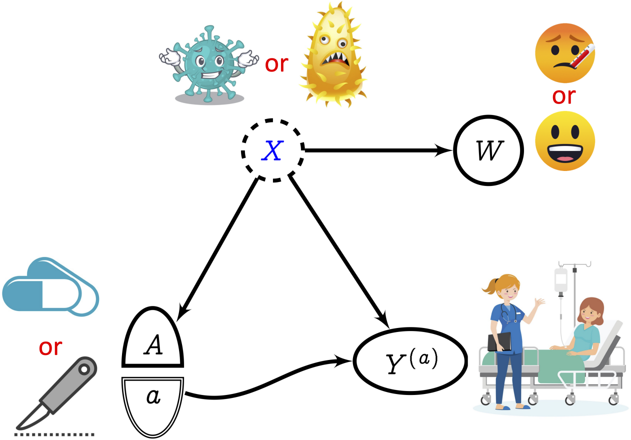
Richardson, Robins (2013), Single World Intervention Graphs (SWIGs): A Unification of the Counterfactual and Graphical Approaches to Causality

We record symptom W, not disease X



- $P(W = \text{fever} | X = \text{mild}) = 0.2$
- $P(W = \text{fever} | X = \text{severe}) = 0.8$

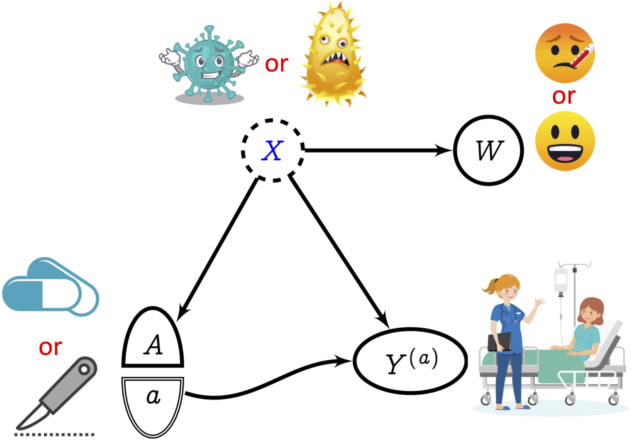
We record symptom W, not disease X



- $P(W = \text{fever} | X = \text{mild}) = 0.2$
- $P(W = \text{fever} | X = \text{severe}) = 0.8$

Could we just write: $P(Y^{(a)}) \overset{?}{=} \sum_{w\in\{0,1\}} \mathbb{E}[Y|a,w]p(w)$

We record symptom W, not disease X



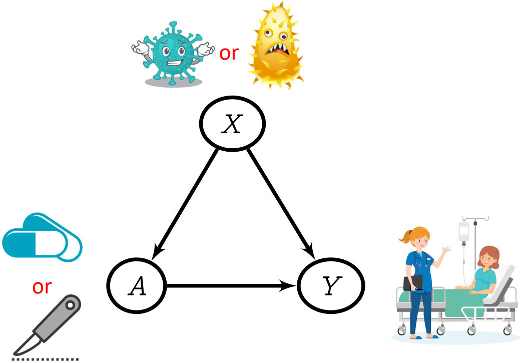
Wrong recommendation made:

- $\sum_{w\in\{0,1\}} \mathbb{E}[\text{cured}|\text{pills}, w]p(w) = 0.8 \quad (\neq 0.64)$
- $\sum_{w\in\{0,1\}} \mathbb{E}[\text{cured}|\text{surgery}, w]p(w) = 0.73 \quad (\neq 0.75)$

Correct answer impossible without observing X

Pearl (2010), On Measurement Bias in Causal Inference

Some core assumptions



Assume:

- Stable Unit Treatment Value Assumption (aka "no interference"),
- Conditional exchangeability $Y^{(a)} \perp\!\!\!\perp A | X$.
- Overlap.

Outline

Causal effect estimation, with hidden covariates X:

- Use proxy variables (negative controls)

Applications: effect of actions under

- privacy constraints (email, ads, DMA)
- data gathering constraints (edge computing)
- fundamental limitations (preferences, state of mind)

Outline

Causal effect estimation, with hidden covariates X:

- Use proxy variables (negative controls)

Applications: effect of actions under

- privacy constraints (email, ads, DMA)
- data gathering constraints (edge computing)
- fundamental limitations (preferences, state of mind)

What's new and why?

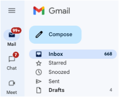
- Treatment A, proxy variables, etc can be multivariate, complicated...
- ...by using adaptive neural net feature representations
- Don't ~~meet your heroes~~ model your hidden variables!

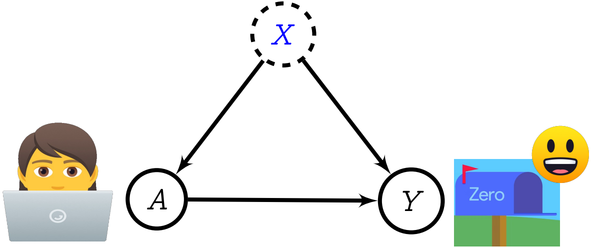
What are proxies, and when are they useful?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: email inbox
- A: prioritize important
- Y: outcome (efficiency)



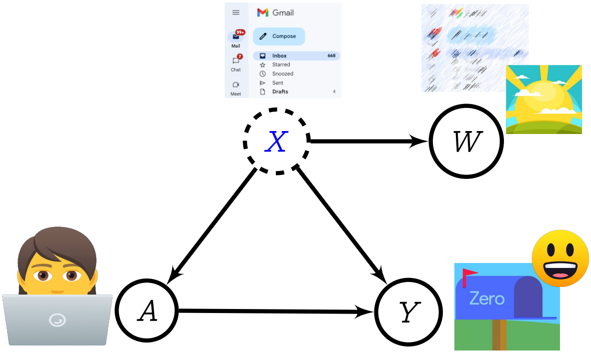


What are proxies, and when are they useful?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: email inbox
- A: prioritize important
- Y: outcome (efficiency)
- W: anonymized inbox before action A

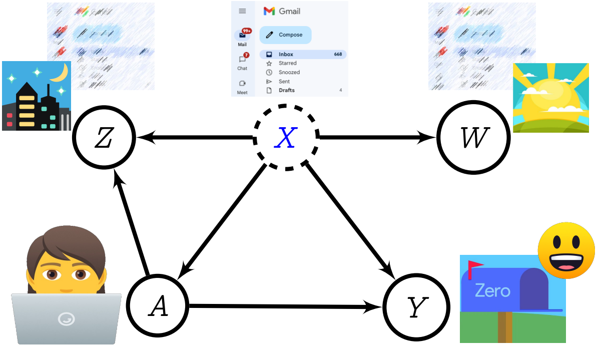


What are proxies, and when are they useful?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: email inbox
- A: prioritize important
- Y: outcome (efficiency)
- W: anonymized inbox before action A
- Z: anonymized inbox after action A

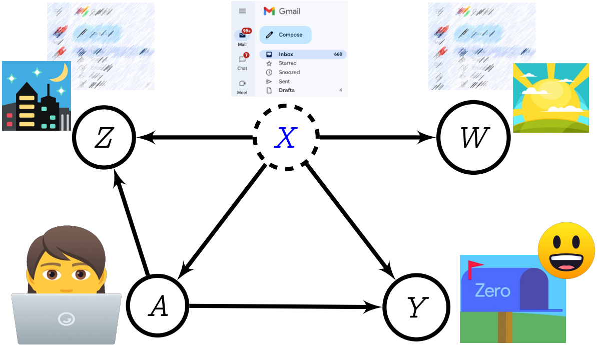


What are proxies, and when are they useful?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: email inbox
- A: prioritize important
- Y: outcome (efficiency)
- W: anonymized inbox before action A
- Z: anonymized inbox after action A



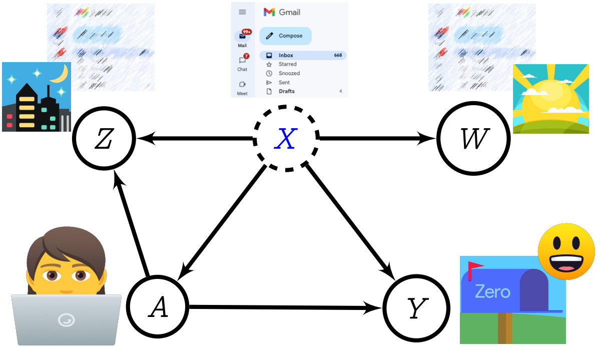
$\Longrightarrow$ Can recover $\mathbb{E}(Y^{(a)})$ from observational data

What are proxies, and when are they useful?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: email inbox
- A: prioritize important
- Y: outcome (efficiency)
- W: anonymized inbox before action A
- Z: anonymized inbox after action A



$\Longrightarrow$ Can recover $\mathbb{E}(Y^{(a)})$ from observational data

$\Longrightarrow$ More usefully: evaluate novel, on-device policy:

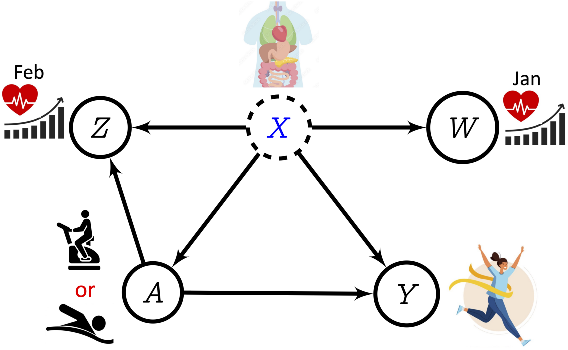
$$\mathbb{E}(Y^{(\pi(A|X))})$$

What are proxies, and when are they useful (2)?

Unobserved X with (possibly) complex nonlinear effects on A, Y

In this example:

- X: true physical status
- A: exercise regimes
- Y: fitness goal
- W: health readings before A
- Z: health readings after A

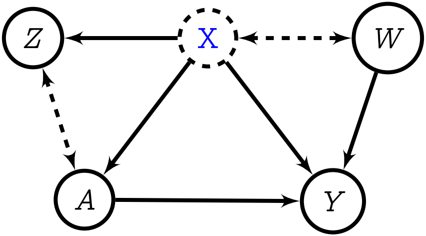


Proxy variables: general setting

Unobserved X with (possibly) complex nonlinear effects on A, Y

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- Z: treatment proxy
- W outcome proxy



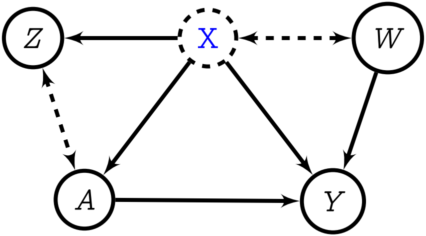
Miao, Geng, Tchetgen Tchetgen (2018): Identifying causal effects with proxy variables of an unmeasured confounder.

Proxy variables: general setting

Unobserved X with (possibly) complex nonlinear effects on A, Y

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- Z: treatment proxy
- W outcome proxy



Structural assumptions:

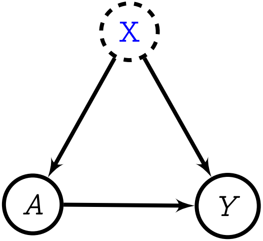
$$W \perp\!\!\!\perp (Z, A) | X$$
$$Y \perp\!\!\!\perp Z | (A, X)$$

Miao, Geng, Tchetgen Tchetgen (2018): Identifying causal effects with proxy variables of an unmeasured confounder.

Why proxy variables? A simple proof

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome



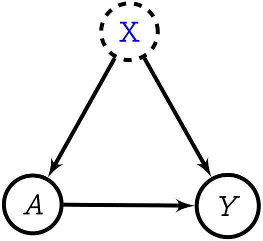
If X were observed,

$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y|x_i, a)P(x_i)$$

Why proxy variables? A simple proof

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome



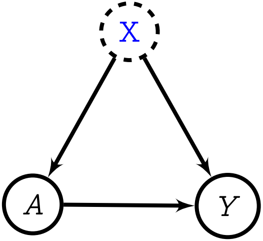
If X were observed,

$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y|x_i, a)P(x_i) = \underbrace{P(Y|X, a)}_{d_y \times d_x}\underbrace{P(X)}_{d_x \times 1}$$

Why proxy variables? A simple proof

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome



If X were observed,

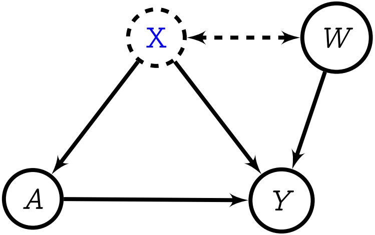
$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y|x_i, a)P(x_i) = \underbrace{P(Y|X, a)}_{d_y \times d_x}\underbrace{P(X)}_{d_x \times 1}$$

Goal: "get rid of the blue" X

...add the outcome proxy W

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- W: outcome proxy



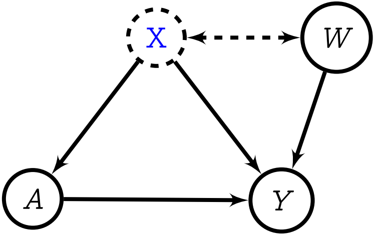
For each a, if we could solve:

$$\underbrace{P(Y|X,a)}_{d_y\times d_x} = \underbrace{H_{w,a}}_{d_y\times d_w}\underbrace{P(W|X)}_{d_w\times d_x}$$

...add the outcome proxy W

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- W: outcome proxy



For each a, if we could solve:

$$\underbrace{P(Y|X,a)}_{d_y\times d_x} = \underbrace{H_{w,a}}_{d_y\times d_w}\underbrace{P(W|X)}_{d_w\times d_x}$$

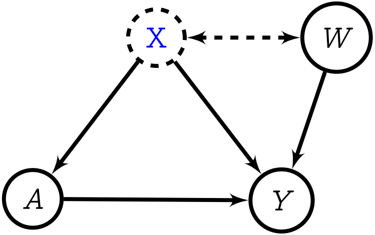
.....then

$$P(Y^{(a)}) = P(Y|X,a)P(X)$$

...add the outcome proxy W

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- W: outcome proxy



For each a, if we could solve:

$$\underbrace{P(Y|X,a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w}\underbrace{P(W|X)}_{d_w \times d_x}$$

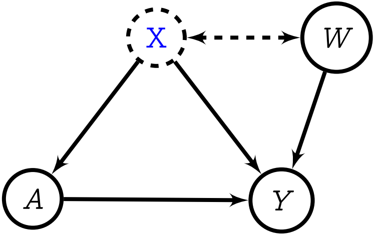
.....then

$$\begin{aligned} P(Y^{(a)}) &= P(Y|X,a)P(X) \\ &= H_{w,a}P(W|X)P(X) \end{aligned}$$

...add the outcome proxy W

The definitions are:

- X: unobserved confounder.
- A: treatment
- Y: outcome
- W: outcome proxy



For each a, if we could solve:

$$\underbrace{P(Y|X,a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w}\underbrace{P(W|X)}_{d_w \times d_x}$$

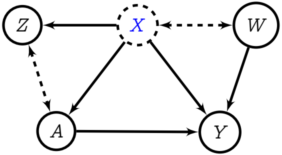
.....then

$$\begin{aligned} P(Y^{(a)}) &= P(Y|X,a)P(X) \\ &= H_{w,a}P(W|X)P(X) \\ &= H_{w,a}P(W) \end{aligned}$$

...now project onto $p(X|Z, a)$

From last slide,

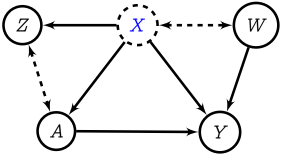
$$P(Y|X, a) = H_{w,a} P(W|X)$$



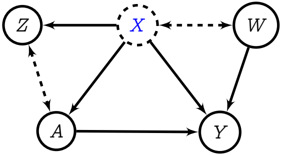
...now project onto $p(X|Z,a)$

From last slide,

$$P(Y|X,a)\underbrace{p(X|Z,a)}_{d_x \times d_z} = H_{w,a} P(W|X)\underbrace{p(X|Z,a)}_{d_x \times d_z}$$



...now project onto $p(X|Z,a)$



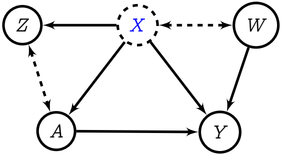
From last slide,

$$P(Y|X,a)\underbrace{p(X|Z,a)}_{d_x \times d_z} = H_{w,a}P(W|X)\underbrace{p(X|Z,a)}_{d_x \times d_z}$$

Because $W \perp\!\!\!\perp (Z,A)|X$,

$$P(W|X)p(X|Z,a) = P(W|Z,a)$$

...now project onto $p(X|Z,a)$



From last slide,

$$P(Y|X,a)\underbrace{p(X|Z,a)}_{d_x \times d_z} = H_{w,a}P(W|X)\underbrace{p(X|Z,a)}_{d_x \times d_z}$$

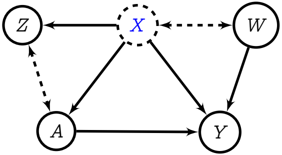
Because $W \perp\!\!\!\perp (Z,A)|X$,

$$P(W|X)p(X|Z,a) = P(W|Z,a)$$

Because $Y \perp\!\!\!\perp Z|(A,X)$,

$$P(Y|X,a)p(X|Z,a) = P(Y|Z,a)$$

...now project onto $p(X|Z,a)$



From last slide,

$$P(Y|X,a)\underbrace{p(X|Z,a)}_{d_x \times d_z} = H_{w,a} P(W|X)\underbrace{p(X|Z,a)}_{d_x \times d_z}$$

Because $W \perp\!\!\!\perp (Z,A)|X$,

$$P(W|X)p(X|Z,a) = P(W|Z,a)$$

Because $Y \perp\!\!\!\perp Z|(A,X)$,

$$P(Y|X,a)p(X|Z,a) = P(Y|Z,a)$$

Solve for $H_{w,a}$:

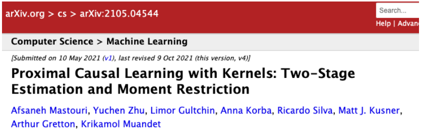
$$P(Y|Z,a) = H_{w,a} P(W|Z,a)$$

Everything observed!

Proxy/Negative Control Methods in the Real World

Unobserved confounders: proxy methods

Kernel features (ICML 2021):



arXiv.org > cs > arXiv:2105.04544

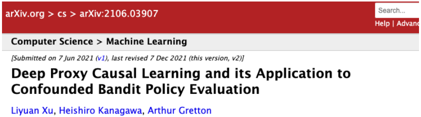
Computer Science > Machine Learning

[Submitted on 10 May 2021 (v1), last revised 9 Oct 2021 (this version, v4)]

Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction

Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt J. Kusner, Arthur Gretton, Krikamol Muandet

NN features (NeurIPS 2021):



arXiv.org > cs > arXiv:2106.03907

Computer Science > Machine Learning

[Submitted on 7 Jun 2021 (v1), last revised 7 Dec 2021 (this version, v2)]

Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation

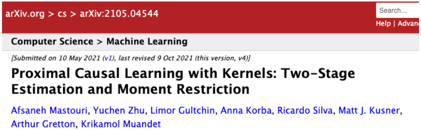
Liyuan Xu, Heishiro Kanagawa, Arthur Gretton

Code for NN and kernel proxy methods:

https://github.com/liyuan9988/DeepFeatureProxyVariable/

Unobserved confounders: proxy methods

Kernel features (ICML 2021):



arXiv.org > cs > arXiv:2105.04544

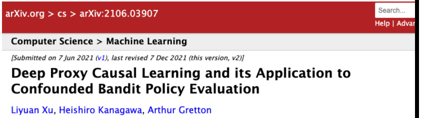
Computer Science > Machine Learning

[Submitted on 10 May 2021 (v1), last revised 9 Oct 2021 (this version, v4)]

Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction

Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt J. Kusner, Arthur Gretton, Krikamol Muandet

NN features (NeurIPS 2021):



arXiv.org > cs > arXiv:2106.03907

Computer Science > Machine Learning

[Submitted on 7 Jun 2021 (v1), last revised 7 Dec 2021 (this version, v2)]

Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation

Liyuan Xu, Heishiro Kanagawa, Arthur Gretton

Code for NN and kernel proxy methods:

`https://github.com/liyuan9988/DeepFeatureProxyVariable/`

One model: linear functions of features

All learned functions will take the form:

$$\gamma(x) = \gamma^\top \varphi_\theta(x)$$

NN approach: Finite dictionaries of learned neural net features $\varphi_\theta(x)$ (linear final layer γ)

Xu, G., A Neural mean embedding approach for back-door and front-door adjustment. (ICLR 23)

Xu, Chen, Srinivasan, de Freitas, Doucet, G. Learning Deep Features in Instrumental Variable Regression. (ICLR 21)

Xu, Kanagawa, G. "Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation". (NeurIPS 21)

Model fitting: *neural* ridge regression

Learn $\gamma_0(x) := \mathbb{E}[\,Y\,|\,X = x]$ from features $\varphi_\theta(x_i)$ with outcomes y_i:

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma \in \mathcal{H}} \left(\sum_{i=1}^{n} \left(y_i - \gamma^\top \varphi_\theta(x_i) \right)^2 + \lambda \|\gamma\|_{\mathcal{H}}^2 \right) \tag{1}$$

Model fitting: *neural* ridge regression

Learn $\gamma_0(x) := \mathbb{E}[Y|X=x]$ from features $\varphi_\theta(x_i)$ with outcomes y_i:

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma \in \mathcal{H}} \left(\sum_{i=1}^{n} \left(y_i - \gamma^\top \varphi_\theta(x_i) \right)^2 + \lambda \|\gamma\|_{\mathcal{H}}^2 \right) \tag{1}$$

Solution for linear final layer γ:

$$\hat{\gamma} = C_{YX}^{(\theta)} (C_{XX}^{(\theta)} + \lambda)^{-1}$$

$$C_{YX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^{n} [y_i \, \varphi_\theta(x_i)^\top]$$

$$C_{XX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^{n} [\varphi_\theta(x_i) \, \varphi_\theta(x_i)^\top]$$

Model fitting: *neural* ridge regression

Learn $\gamma_0(x) := \mathbb{E}[\,Y\,|\,X = x]$ from features $\varphi_\theta(x_i)$ with outcomes y_i:

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma\in\mathcal{H}} \left(\sum_{i=1}^{n} \left(y_i - \gamma^\top \varphi_\theta(x_i)\right)^2 + \lambda\|\gamma\|^2_{\mathcal{H}}\right) \tag{1}$$

Solution for linear final layer γ:

$$\hat{\gamma} = C^{(\theta)}_{YX}(C^{(\theta)}_{XX} + \lambda)^{-1}$$

$$C^{(\theta)}_{YX} = \frac{1}{n}\sum_{i=1}^{n}[y_i\,\varphi_\theta(x_i)^\top]$$

$$C^{(\theta)}_{XX} = \frac{1}{n}\sum_{i=1}^{n}[\varphi_\theta(x_i)\,\varphi_\theta(x_i)^\top]$$

How to solve for θ:

Substitute $\hat{\gamma}$ into (1), backprop through Cholesky for θ.

Model fitting: *neural* ridge regression

Learn $\gamma_0(x) := \mathbb{E}[Y|X=x]$ from features $\varphi_\theta(x_i)$ with outcomes y_i:

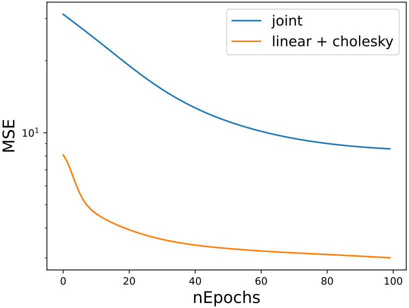
$$\hat{\gamma} \;=\; \arg\min_{\gamma\in\mathcal{H}} \left(\sum_{i=1}^{n}\left(y_i - \gamma^\top\varphi_\theta(x_i)\right)^2 + \lambda\|\gamma\|^2_{\mathcal{H}}\right) \qquad (1)$$

Solution for linear final layer γ:

$$\hat{\gamma} = C^{(\theta)}_{YX}(C^{(\theta)}_{XX}+\lambda)^{-1}$$

$$C^{(\theta)}_{YX} = \frac{1}{n}\sum_{i=1}^{n}[y_i\,\varphi_\theta(x_i)^\top]$$

$$C^{(\theta)}_{XX} = \frac{1}{n}\sum_{i=1}^{n}[\varphi_\theta(x_i)\,\varphi_\theta(x_i)^\top]$$



MNIST, 4 layer FF, sigmoid, fully connected

How to solve for θ:

Substitute $\hat{\gamma}$ into (1), backprop through Cholesky for θ.

Road map: NN proxy learning

We'll proceed as follows:

- Proxy relation for continuous variables
- Loss function for deep proxy learning
- Define primary (ridge) regression with this loss
- Define secondary (ridge) regression as input to primary

Proxy relation, general domains

If X were observed, we would write (average treatment effect)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a,x)p(x)dx.$$

....but we do not observe X.

Proxy relation, general domains

If X were observed, we would write (average treatment effect)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a,x)p(x)dx.$$

....but we do not observe X.

Main theorem: Assume we solved for link function:

$$\mathbb{E}(Y|a,z) = \mathbb{E}_{W|a,z} h_y(W,a)$$

- "Primary" $\mathbb{E}(Y|a,z)$, "secondary" $\mathbb{E}_{W|a,z}$ linked by h_y
- All variables observed, X not seen *or modeled.*

(Fredholm equation of first kind: existence of solution requires identifiability conditions)

Proxy relation, general domains

If X were observed, we would write (average treatment effect)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a,x)p(x)dx.$$

....but we do not observe X.

Main theorem: Assume we solved for link function:

$$\mathbb{E}(Y|a,z) = \mathbb{E}_{W|a,z} h_y(W,a)$$

- "Primary" $\mathbb{E}(Y|a,z)$, "secondary" $\mathbb{E}_{W|a,z}$ linked by h_y
- All variables observed, X not seen *or modeled.*

Average treatment effect via $p(w)$:

$$\mathbb{E}(Y^{(a)}) = \int_w h_y(a,w)p(w)dw$$

(Fredholm equation of first kind: existence of solution requires identifiability conditions)

Proxy relation, general domains

If X were observed, we would write (average treatment effect)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a,x)p(x)dx.$$

....but we do not observe X.

Main theorem: Assume we solved for link function:

$$\mathbb{E}(Y|a,z) = \mathbb{E}_{W|a,z} h_y(W,a)$$

- "Primary" $\mathbb{E}(Y|a,z)$, "secondary" $\mathbb{E}_{W|a,z}$ linked by h_y
- All variables observed, X not seen *or modeled.*

Average treatment effect via $p(w)$:

$$\mathbb{E}(Y^{(a)}) = \int_w h_y(a,w)p(w)dw$$

Challenge: need a loss function for h_y

(Fredholm equation of first kind: existence of solution requires identifiability conditions)

Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2$$

Why?

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a,z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2$$

Why?

$f^*(a, z) = \mathbb{E}(Y|a, z)$ solves

$$\operatorname*{argmin}_{f} \mathbb{E}_{Y,A,Z} (Y - f(A, Z))^2$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2$$

Why?

$f^*(a, z) = \mathbb{E}(Y|a, z)$ solves

$$\underset{f}{\operatorname{argmin}} \, \mathbb{E}_{Y,A,Z} (Y - f(A, Z))^2$$

...and by the proxy model above,

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN for link $h_y(a, w)$

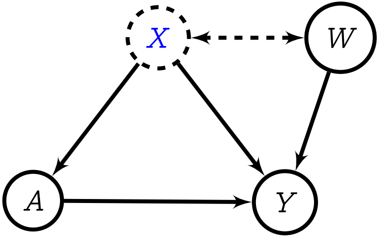
The link function is a function of two arguments

$$h_y(a, w) = \gamma^\top \left[\varphi_\theta(w) \otimes \varphi_\xi(a)\right] = \gamma^\top \begin{bmatrix} \varphi_{\theta,1}(w)\varphi_{\xi,1}(a) \\ \varphi_{\theta,1}(w)\varphi_{\xi,2}(a) \\ \vdots \\ \varphi_{\theta,2}(w)\varphi_{\xi,1}(a) \\ \vdots \end{bmatrix}$$

Assume we have:

- output proxy NN features $\varphi_\theta(w)$
- treatment NN features $\varphi_\xi(a)$
- linear final layer γ

(argument of feature map indicates feature space)



NN for link $h_y(a, w)$

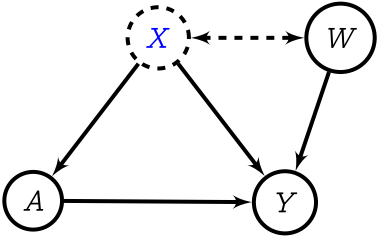
The link function is a function of two arguments

$$h_y(a, w) = \gamma^\top [\varphi_\theta(w) \otimes \varphi_\xi(a)]$$

Assume we have:

- output proxy NN features $\varphi_\theta(w)$
- treatment NN features $\varphi_\xi(a)$
- linear final layer γ

(argument of feature map indicates feature space)



Questions:

- Why feature map $\varphi_\theta(w) \otimes \varphi_\xi(a)$?
- Why final linear layer γ?

Both are necessary (next slide)!

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

How to get conditional expectation $\mathbb{E}_{W|a,z} h_y(W, a)$?

Density estimation for $p(W|a, z)$? Sample from $p(W|a, z)$?

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$h_y(W, a) = \qquad \left[\gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right]$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\mathbb{E}_{W|a,z}\ h_y(W, a) = \ \mathbb{E}_{W|a,z}\ \left[\gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a))\right]$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z} \left(Y - \mathbb{E}_{W|A,Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\begin{aligned} \mathbb{E}_{W|a,z}\, h_y(W, a) &= \mathbb{E}_{W|a,z} \left[\gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right] \\ &= \gamma^\top \Big(\underbrace{\mathbb{E}_{W|a,z} [\varphi_\theta(W)]}_{\text{cond. feat. mean}} \otimes \varphi_\xi(a) \Big) \end{aligned}$$

(this is why linear γ and feature map $\varphi_\theta(w) \otimes \varphi_\xi(a)$)

Deaner (2021).
Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).
Xu, Kanagawa, G. (2021).

NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a,z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg\min_{h_y} \mathbb{E}_{Y,A,Z}\left(Y - \mathbb{E}_{W|A,Z} h_y(W, A)\right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\begin{aligned}\mathbb{E}_{W|a,z}\, h_y(W, a) &= \mathbb{E}_{W|a,z}\left[\gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a))\right] \\ &= \gamma^\top \Big(\underbrace{\mathbb{E}_{W|a,z}[\varphi_\theta(W)]}_{\text{cond. feat. mean}} \otimes \varphi_\xi(a)\Big)\end{aligned}$$

Ridge regression (again!)

$$\mathbb{E}_{W|a,z}\varphi_\theta(W) = \hat{F}_{\theta,\zeta}\varphi_\zeta(a, z)$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

NN ridge regression for $\mathbb{E}_{W|a,z}\varphi_\theta(W)$

Secondary regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}_{W|a,z}\varphi_\theta(W) = \hat{F}_{\theta,\zeta}\varphi_\zeta(a,z)$$

with RR loss

$$\mathbb{E}_{W,A,Z}\,\|\varphi_\theta(W) - F\varphi_\zeta(A,Z)\|^2 + \lambda_1\|F\|^2$$

$\hat{F}_{\theta,\zeta}$ in closed form wrt ϕ_θ, ϕ_ζ.

Xu, Kanagawa, G. (2021).

NN ridge regression for $\mathbb{E}_{W|a,z}\varphi_\theta(W)$

Secondary regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}_{W|a,z}\varphi_\theta(W) = \hat{F}_{\theta,\zeta}\varphi_\zeta(a,z)$$

with RR loss

$$\mathbb{E}_{W,A,Z}\left\|\varphi_\theta(W) - F\varphi_\zeta(A,Z)\right\|^2 + \lambda_1\|F\|^2$$

$\hat{F}_{\theta,\zeta}$ in closed form wrt ϕ_θ, ϕ_ζ.

Plug $\hat{F}_{\theta,\zeta}$ into S1 loss, backprop through Cholesky for ζ (...not θ...why not?)

Xu, Kanagawa, G. (2021).

Final algorithm

Solve for θ, ξ, ζ:

Repeat until convergence:

- Secondary: Solve for $\hat{F}_{\theta,\zeta}$, then gradient steps on ζ (backprop through Cholesky)

Xu, Kanagawa, G. (2021).

Final algorithm

Solve for θ, ξ, ζ:

Repeat until convergence:

- Secondary: Solve for $\hat{F}_{\theta,\zeta}$, then gradient steps on ζ (backprop through Cholesky)
- Primary: Solve for $\hat{\gamma}$ in terms of $\hat{F}_{\theta,\zeta}\phi_{\zeta}(A, Z)$ and $\varphi_{\xi}(A)$

Xu, Kanagawa, G. (2021).

Final algorithm

Solve for θ, ξ, ζ:

Repeat until convergence:

- Secondary: Solve for $\hat{F}_{\theta,\zeta}$, then gradient steps on ζ (backprop through Cholesky)
- Primary: Solve for $\hat{\gamma}$ in terms of $\hat{F}_{\theta,\zeta}\phi_{\zeta}(A, Z)$ and $\varphi_{\xi}(A)$
- Primary: Gradient steps on θ, ξ (backprop through Cholesky)
    - $\hat{F}_{\theta,\zeta}$ remains optimal wrt current φ_{θ}.

Xu, Kanagawa, G. (2021).

Final algorithm

Solve for θ, ξ, ζ:

Repeat until convergence:

- Secondary: Solve for $\hat{F}_{\theta,\zeta}$, then gradient steps on ζ (backprop through Cholesky)
- Primary: Solve for $\hat{\gamma}$ in terms of $\hat{F}_{\theta,\zeta}\phi_{\zeta}(A, Z)$ and $\varphi_{\xi}(A)$
- Primary: Gradient steps on θ, ξ (backprop through Cholesky)
    - $\hat{F}_{\theta,\zeta}$ remains optimal wrt current φ_{θ}.

Iterate between updates of θ, ξ and ζ

Xu, Kanagawa, G. (2021).

Final algorithm

Solve for θ, ξ, ζ:

Repeat until convergence:

- Secondary: Solve for $\hat{F}_{\theta,\zeta}$, then gradient steps on ζ (backprop through Cholesky)
- Primary: Solve for $\hat{\gamma}$ in terms of $\hat{F}_{\theta,\zeta}\phi_{\zeta}(A, Z)$ and $\varphi_{\xi}(A)$
- Primary: Gradient steps on θ, ξ (backprop through Cholesky)
  - $\hat{F}_{\theta,\zeta}$ remains optimal wrt current φ_{θ}.

Iterate between updates of θ, ξ and ζ

Key point: features $\varphi_{\theta}(W)$ learned specially for:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

Contrast with autoencoders/sampling: must reconstruct/sample all of W.

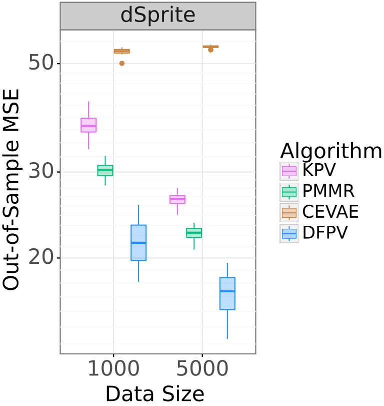
Xu, Kanagawa, G. (2021).

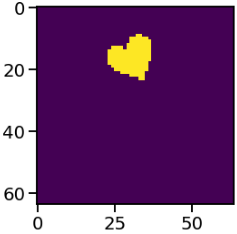
Experiments

Synthetic experiment, adaptive neural net features

dSprite example:

- $X = \{\texttt{scale}, \texttt{rotation}, \texttt{posX}, \texttt{posY}\}$
- Treatment A is the image generated (with Gaussian noise)
- Outcome Y is quadratic function of A with multiplicative confounding by posY.
- $Z = \{\texttt{scale}, \texttt{rotation}, \texttt{posX}\}$,
  $W =$ noisy image sharing posY
- Comparison with CEVAE (Louzios et al. 2017)



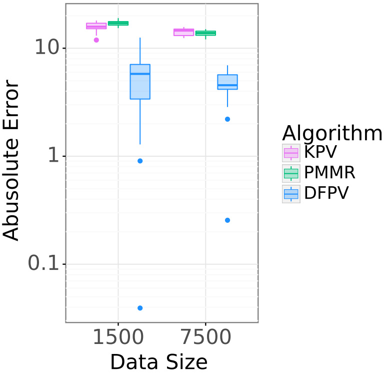


Louizos, Shalit, Mooij, Sontag, Zemel, Welling, Causal Effect Inference with Deep Latent-Variable Models (2017)

Confounded offline policy evaluation

Synthetic dataset, demand prediction for flight purchase.

- Treatment A is ticket price.
- Policy $A \sim \pi(Z)$ depends on fuel price.

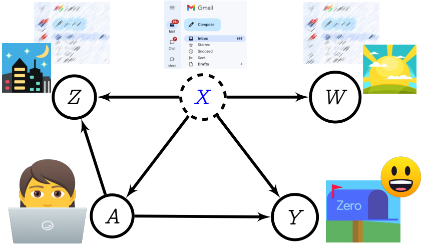


Conclusion

Causal effect estimation with unobserved X, (possibly) complex nonlinear effects on A, Y

We need to observe:

- Treatment proxy Z (interacts with A, but not directly with Y)
- Outcome proxy W (no direct interaction with A, can affect Y)

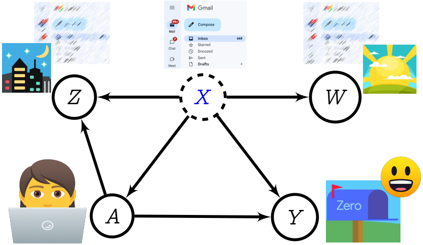


Conclusion

Causal effect estimation with unobserved X, (possibly) complex nonlinear effects on A, Y

We need to observe:

- Treatment proxy Z (interacts with A, but not directly with Y)
- Outcome proxy W (no direct interaction with A, can affect Y)



Key messages:

- Don't ~~meet your heroes~~ model/sample latents X
- Don't model all of W, only relevant features for Y
- "Ridge regression is all you need"

Code available:

https://github.com/liyuan9988/DeepFeatureProxyVariable/

Research support

Work supported by:

The Gatsby Charitable Foundation

GATSBY

Google Deepmind

Google DeepMind

Questions?

THE KERNEL BREWERY LONDON
BALTIC PORTER
7.3% ABV
THE KERNEL
EXPORT STOUT
LONDON 1890
7.8% ABV