Causal Effect Estimation with Context and Confounders

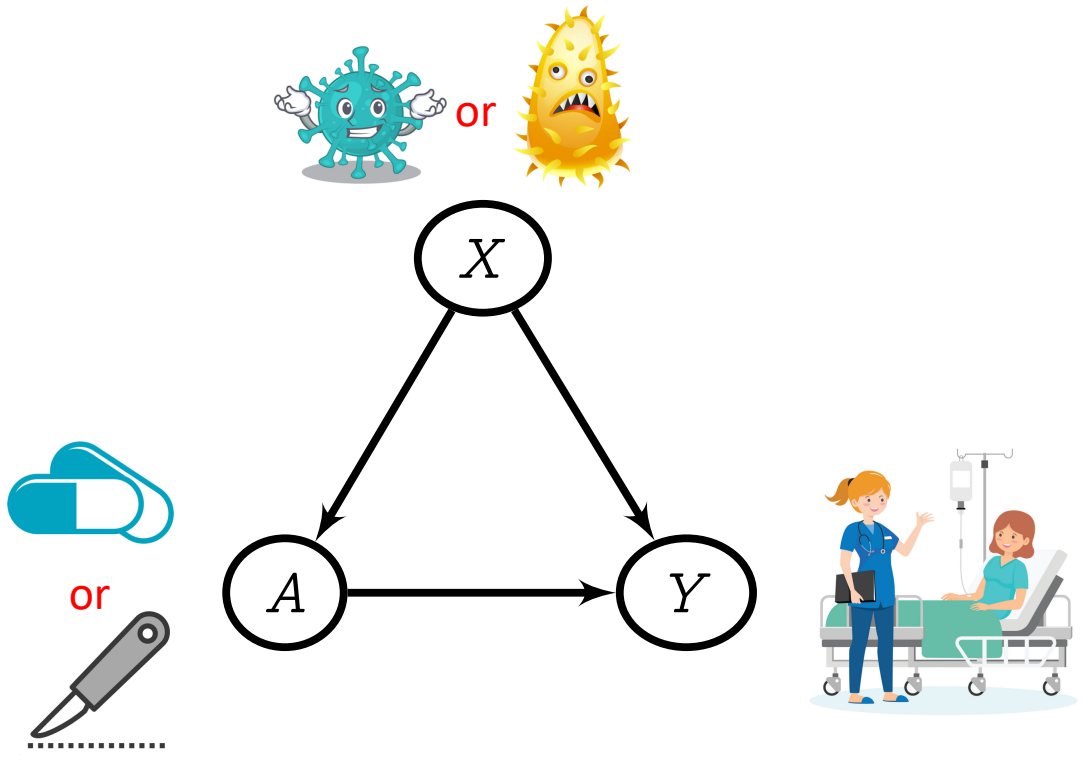
Arthur Gretton

Gatsby Computational Neuroscience Unit, UCL
Google Deepmind

PAISS, Grenoble 2025

Observation vs intervention

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

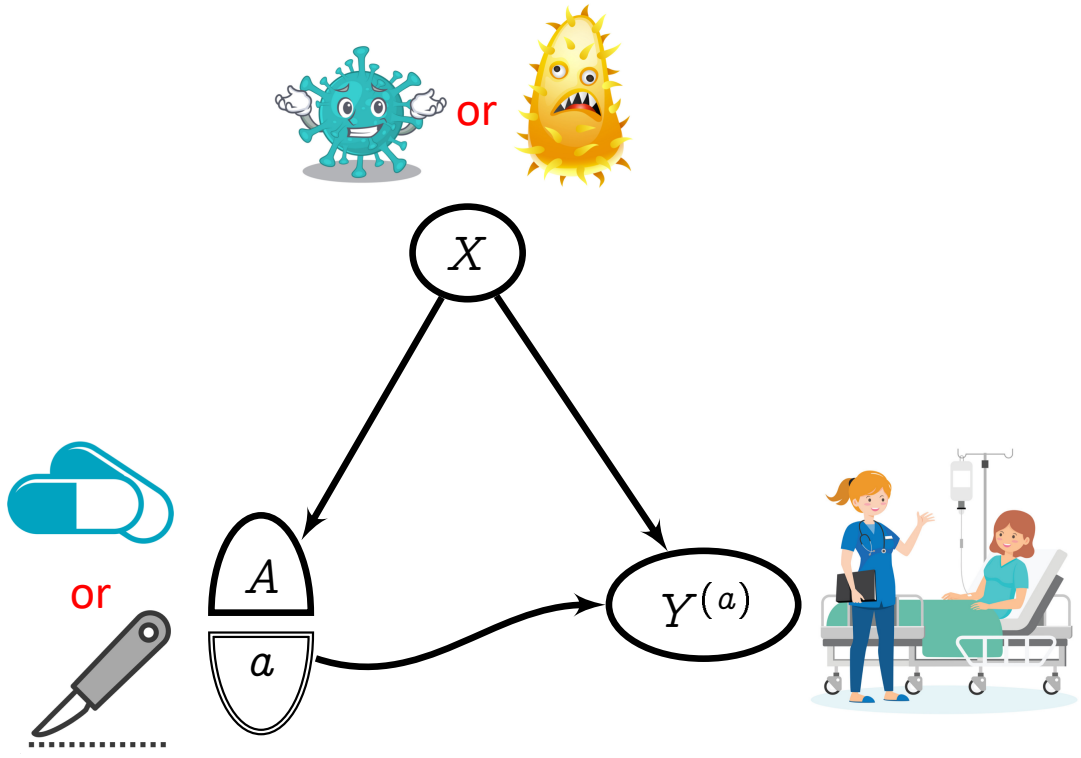


From our *observations* of historical hospital data:

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

Observation vs intervention

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

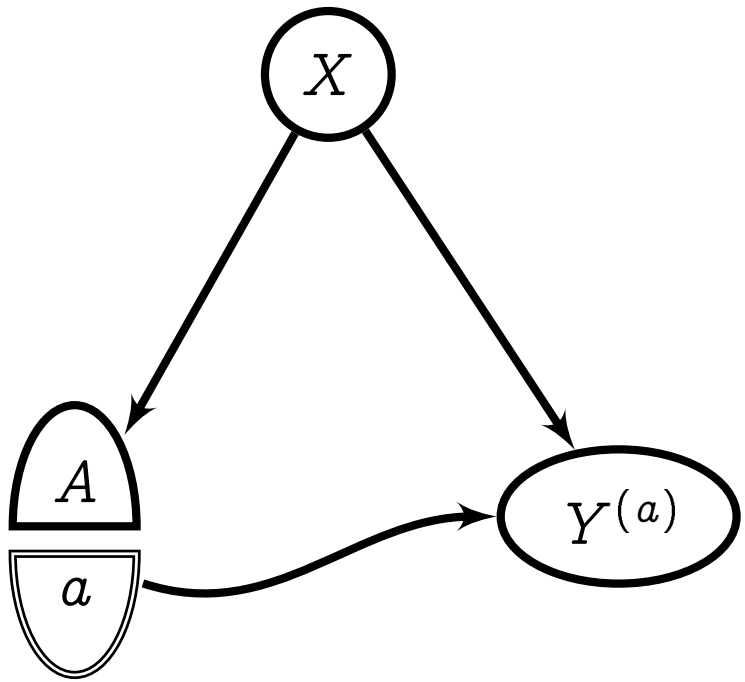


From our *intervention* (making all patients take a treatment):

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

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

Questions we will solve



Outline

Causal effect estimation, observed covariates:

- Average treatment effect (ATE), *conditional* average treatment effect (CATE)

Causal effect estimation, hidden covariates:

- ... instrumental variables, proxy variables

What's new? What is it good for?

- Treatment A, covariates X, etc can be multivariate, complicated...
- ...by using kernel or adaptive neural net feature representations

Model assumption: linear functions of features

All learned functions will take the form:

$$\gamma(x) = \gamma^\top \varphi(x) = \langle \gamma, \varphi(x) \rangle_{\mathcal{H}}$$

Model assumption: linear functions of features

All learned functions will take the form:

$$\gamma(x) = \gamma^\top \varphi(x) = \langle \gamma, \varphi(x) \rangle_{\mathcal{H}}$$

Option 1: 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)

Option 2: Infinite dictionaries of fixed kernel features:

$$\langle \varphi(x_i), \varphi(x) \rangle_{\mathcal{H}} = k(x_i, x)$$

Kernel is feature dot product.

Singh, Xu, G. Kernel Methods for Causal Functions: Dose, Heterogeneous, and Incremental Response Curves. (Biometrika, 2023)

Singh, Sahani, G. Kernel Instrumental Variable Regression. (NeurIPS 19)

Model fitting: *neural* ridge regression

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

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

Model fitting: *neural* ridge regression

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

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

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:

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

Solution for linear final layer γ:

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

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

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

How to solve for θ:

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

Model fitting: *kernel* ridge regression

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

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma \in \mathcal{H}} \left(\sum_{i=1}^{n} (y_i - \langle \gamma, \varphi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \|\gamma\|_{\mathcal{H}}^2 \right).$$

Model fitting: *kernel* ridge regression

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

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma \in \mathcal{H}} \left(\sum_{i=1}^{n} (y_i - \langle \gamma, \varphi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \|\gamma\|_{\mathcal{H}}^2 \right).$$

Infinite dimensional solution at x:

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

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

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

Model fitting: *kernel* ridge regression

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

$$\hat{\gamma} \quad = \quad \arg\min_{\gamma \in \mathcal{H}} \left(\sum_{i=1}^{n} (y_i - \langle \gamma, \varphi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \|\gamma\|_{\mathcal{H}}^2 \right).$$

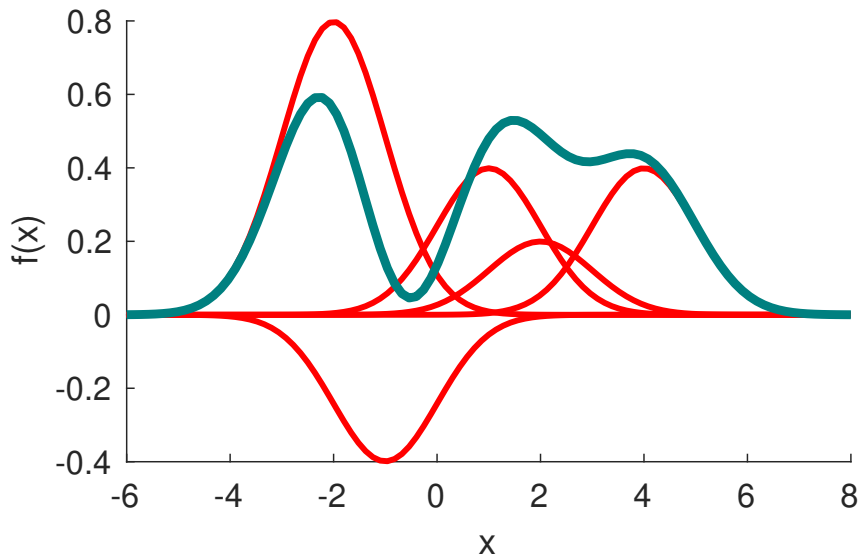
Kernel solution at x
(as weighted sum of y)

$$\hat{\gamma}(x) = \sum_{i=1}^{n} y_i \beta_i(x)$$

$$\beta(x) = (K_{XX} + \lambda I)^{-1} k_{Xx}$$

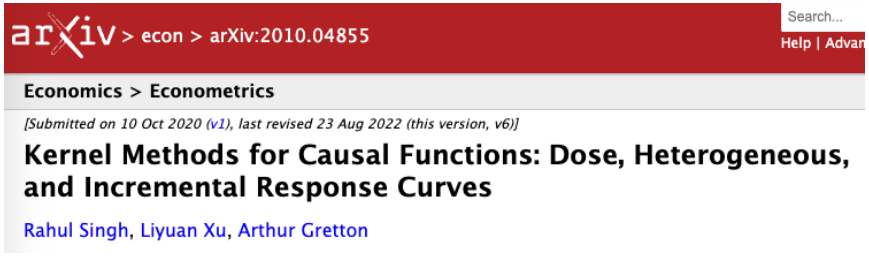
$$(K_{XX})_{ij} = k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}}$$

$$(k_{Xx})_i = k(x_i, x)$$



Observed covariates: (conditional) ATE

Kernels (Biometrika 2023):



NN features (ICLR 2023):

Code for NN and kernel causal estimation with observed covariates:

https://github.com/liyuan9988/DeepFrontBackDoor/

Observed covariates: (conditional) ATE

Kernel features
(in revision, Biometrika):

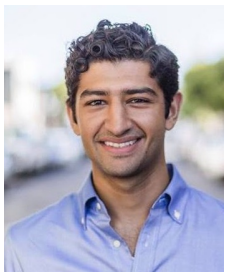
arXiv > econ > arXiv:2010.04855 Search... Help | Adva

Economics > Econometrics

[Submitted on 10 Oct 2020 (v1), last revised 23 Aug 2022 (this version, v6)]

Kernel Methods for Causal Functions: Dose, Heterogeneous, and Incremental Response Curves

Rahul Singh, Liyuan Xu, Arthur Gretton



NN features (ICLR 2023):

arXiv > cs > arXiv:2210.06610 Search... Help | Advan

Computer Science > Machine Learning

[Submitted on 12 Oct 2022]

A Neural Mean Embedding Approach for Back-door and Front-door Adjustment

Liyuan Xu, Arthur Gretton

Code for NN and kernel causal estimation with observed covariates:

https://github.com/liyuan9988/DeepFrontBackDoor/

Average treatment effect

Potential outcome (intervention):

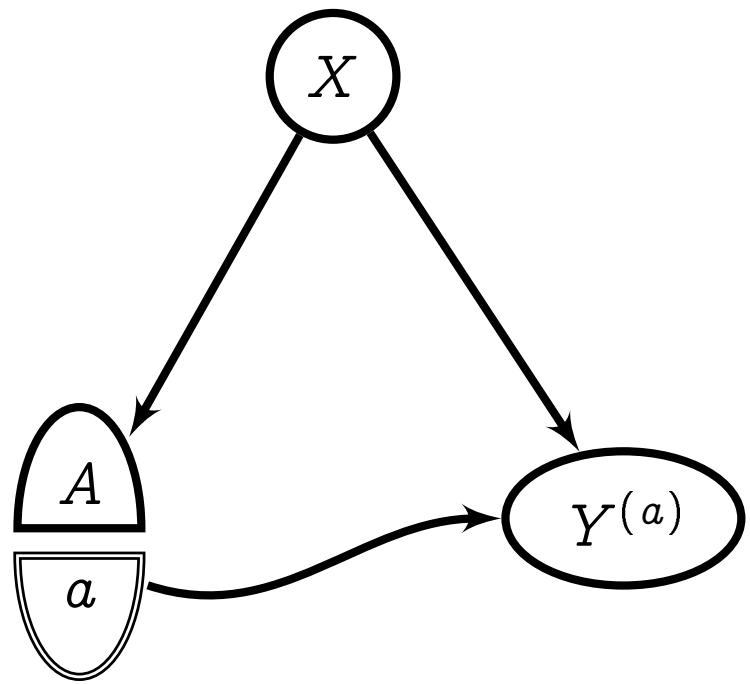
$$\mathbb{E}[Y^{(a)}] = \int \mathbb{E}[Y|a,x]\,dp(x)$$

(the average structural function; in epidemiology, for continuous a, the dose-response curve).

Assume: (1) Stable Unit Treatment Value Assumption (aka "no interference"), (2) Conditional exchangeability $Y^{(a)} \perp\!\!\!\perp A|X$. (3) Overlap.

Example: US job corps, training for disadvantaged youths:

- A: treatment (training hours)
- Y: outcome (percentage employment)
- X: covariates (age, education, marital status, ...)



Multiple inputs via products of kernels

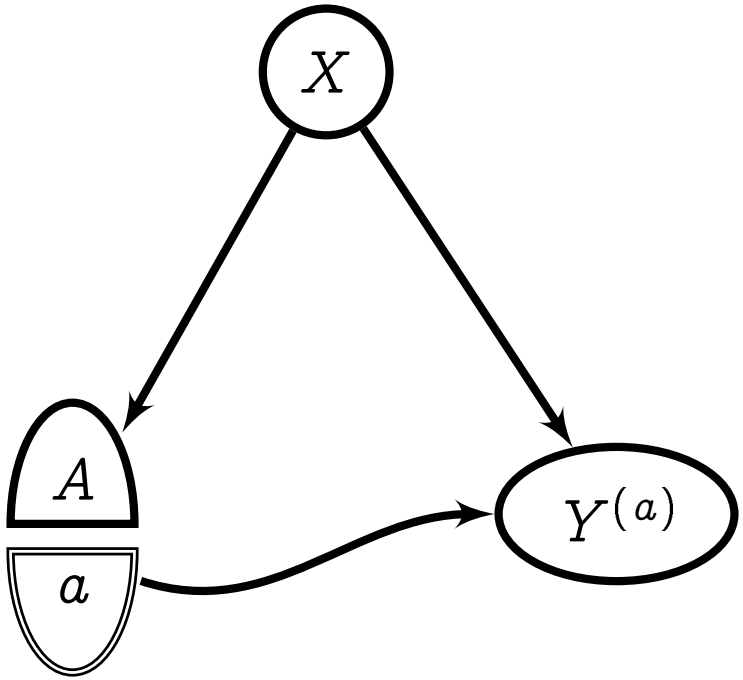
We may predict expected outcome from two inputs

$$\gamma_0(a, x) := \mathbb{E}[\, Y \,|\, a, x]$$

Assume we have:

- covariate features $\varphi(x)$ with kernel $k(x, x')$
- treatment features $\varphi(a)$ with kernel $k(a, a')$

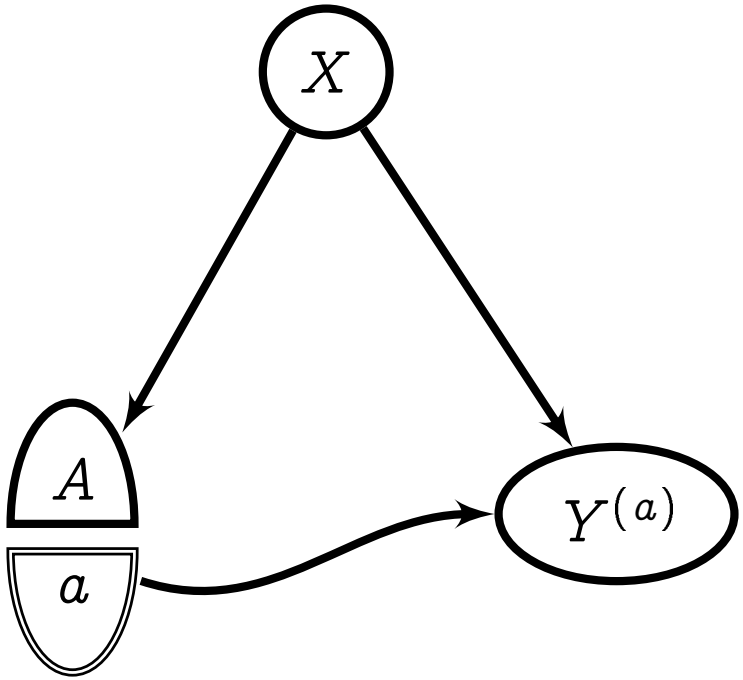
(argument of kernel/feature map indicates feature space)



Multiple inputs via products of kernels

We may predict expected outcome from two inputs

$$\gamma_0(a, x) := \mathbb{E}[\, Y \,|\, a, x]$$



Assume we have:

- covariate features $\varphi(x)$ with kernel $k(x, x')$
- treatment features $\varphi(a)$ with kernel $k(a, a')$

(argument of kernel/feature map indicates feature space)

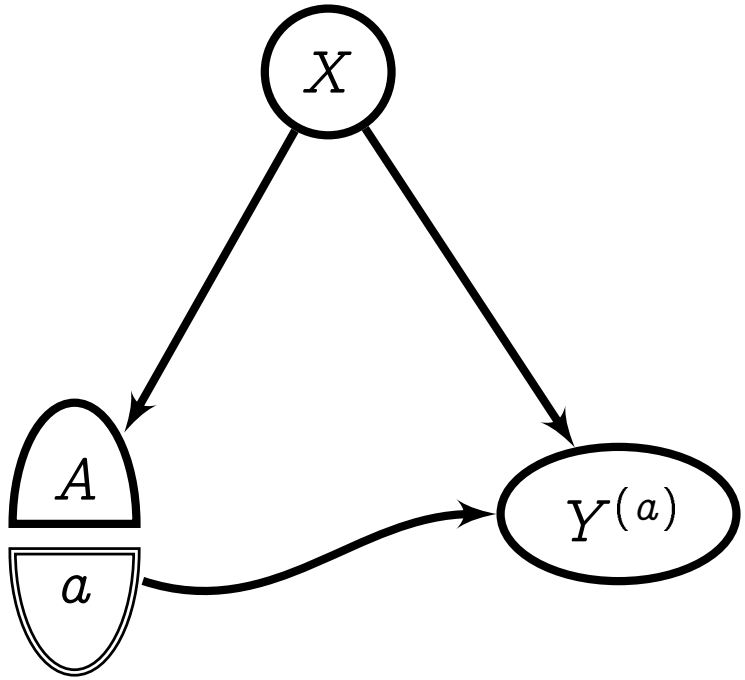
We use outer product of features ($\implies$ product of kernels):

$$\phi(x, a) = \varphi(a) \otimes \varphi(x) \qquad \mathfrak{K}([a, x], [a', x']) = k(a, a')k(x, x')$$

Multiple inputs via products of kernels

We may predict expected outcome from two inputs

$$\gamma_0(a, x) := \mathbb{E}[\, Y \,|\, a, x]$$



Assume we have:

- covariate features $\varphi(x)$ with kernel $k(x, x')$
- treatment features $\varphi(a)$ with kernel $k(a, a')$

(argument of kernel/feature map indicates feature space)

We use outer product of features ($\implies$ product of kernels):

$$\phi(x, a) = \varphi(a) \otimes \varphi(x) \qquad \mathfrak{K}([a, x], [a', x']) = k(a, a')k(x, x')$$

Ridge regression solution:

$$\hat{\gamma}(x, a) = \sum_{i=1}^{n} y_i \beta_i(a, x), \quad \beta(a, x) = [K_{AA} \odot K_{XX} + \lambda I]^{-1} K_{Aa} \odot K_{Xx}$$

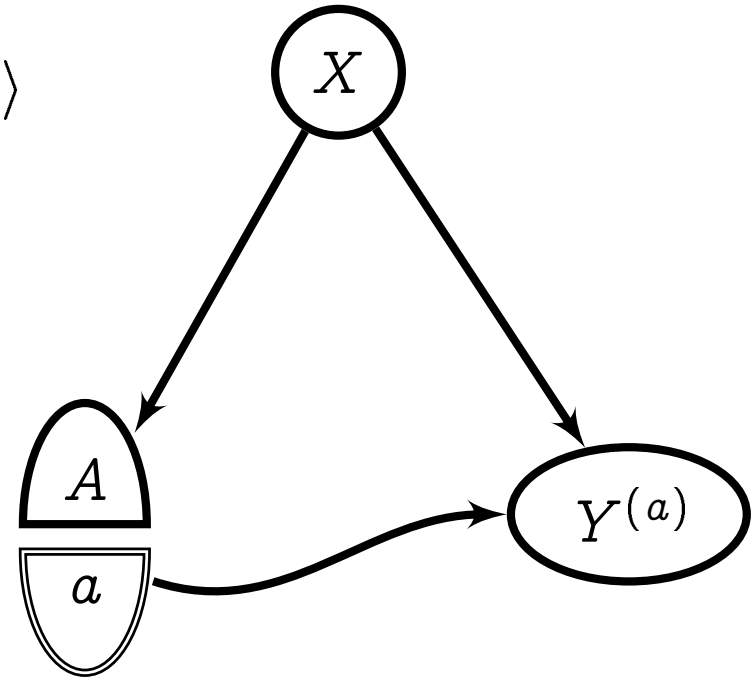
ATE (dose-response curve)

Well-specified setting:

$$\mathbb{E}[\,Y\,|\,a,x] =: \gamma_0(a,x) = \langle \gamma_0, \varphi(a) \otimes \varphi(x) \rangle$$

ATE as feature space dot product:

$$\begin{aligned} \mathrm{ATE}(a) &= \mathbb{E}[\gamma_0(a,X)] \\ &= \mathbb{E}\left[\langle \gamma_0, \varphi(a) \otimes \varphi(X) \rangle\right] \end{aligned}$$



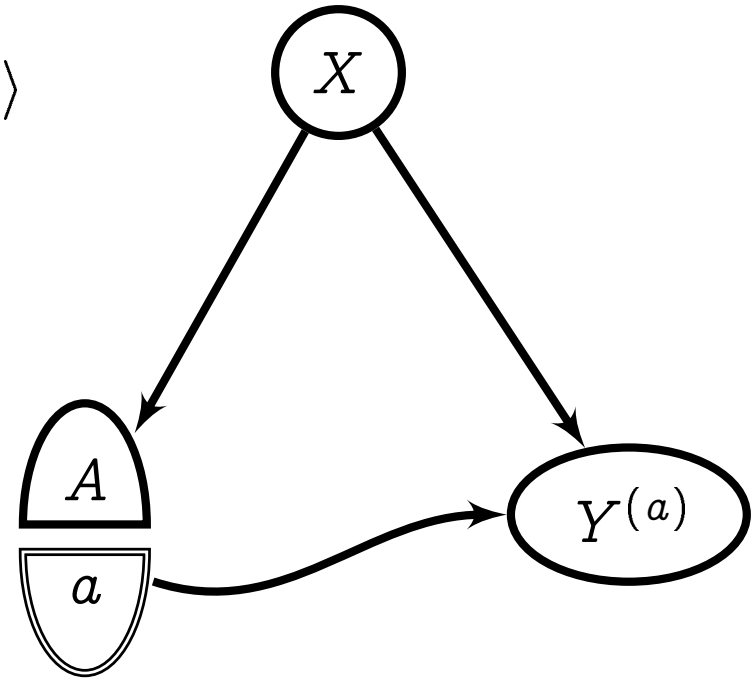
ATE (dose-response curve)

Well-specified setting:

$$\mathbb{E}[\,Y\,|\,a,x\,] =: \gamma_0(a,x) = \langle \gamma_0, \varphi(a) \otimes \varphi(x) \rangle$$

ATE as feature space dot product:

$$\begin{aligned} \mathrm{ATE}(a) &= \mathbb{E}[\gamma_0(a,X)] \\ &= \mathbb{E}\left[\langle \gamma_0, \varphi(a) \otimes \varphi(X) \rangle\right] \\ &= \langle \gamma_0, \varphi(a) \otimes \underbrace{\mu_X}_{\mathbb{E}[\varphi(X)]} \rangle \end{aligned}$$



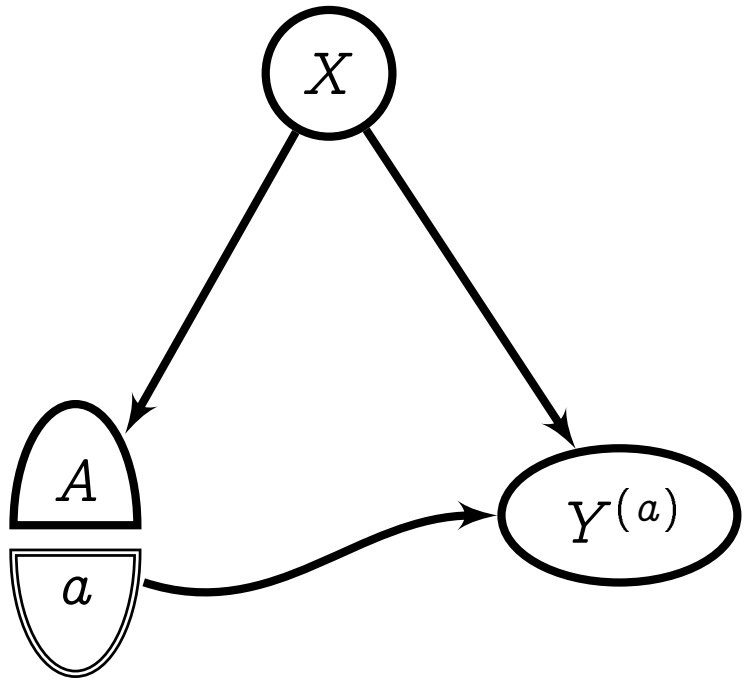
Feature map of probability $P(X)$,

$$\mu_X = [\ldots \mathbb{E}\left[\varphi_i(X)\right] \ldots]$$

ATE: example

US job corps: training for disadvantaged youths:

- X: covariate/context (age, education, marital status, ...)
- A: treatment (training hours)
- Y: outcome (percent employment)



Empirical ATE:

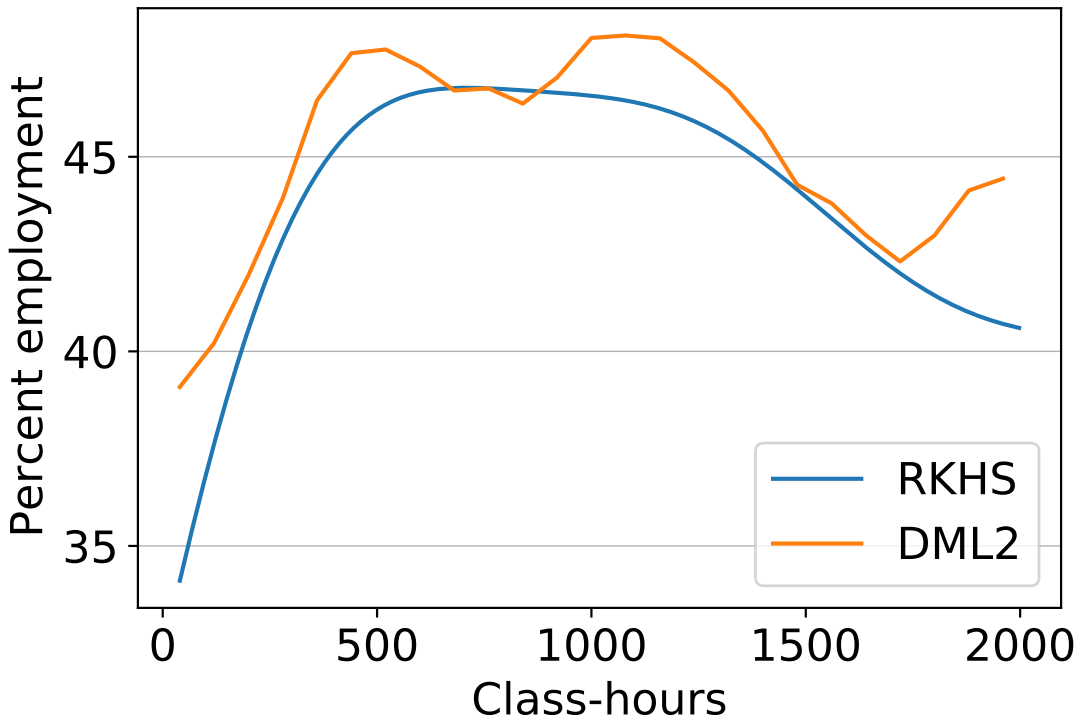
$$\widehat{\mathrm{ATE}}(a) = \widehat{\mathbb{E}}\left[\langle \hat{\gamma}_0, \varphi(X) \otimes \varphi(a) \rangle\right]$$

$$= \frac{1}{n} \sum_{i=1}^{n} Y^\top (K_{AA} \odot K_{XX} + n\lambda I)^{-1} (K_{Aa} \odot K_{Xx_i})$$

Schochet, Burghardt, and McConnell (2008). Does Job Corps work? Impact findings from the national Job Corps study.

Singh, Xu, G (2023).

ATE: results



- First 12.5 weeks of classes confer employment gain: from 35% to 47%.
- [RKHS] is our $\widehat{\text{ATE}}(a)$.
- [DML2] Colangelo, Lee (2020), Double debiased machine learning nonparametric inference with continuous treatments.

Singh, Xu, G (2023)

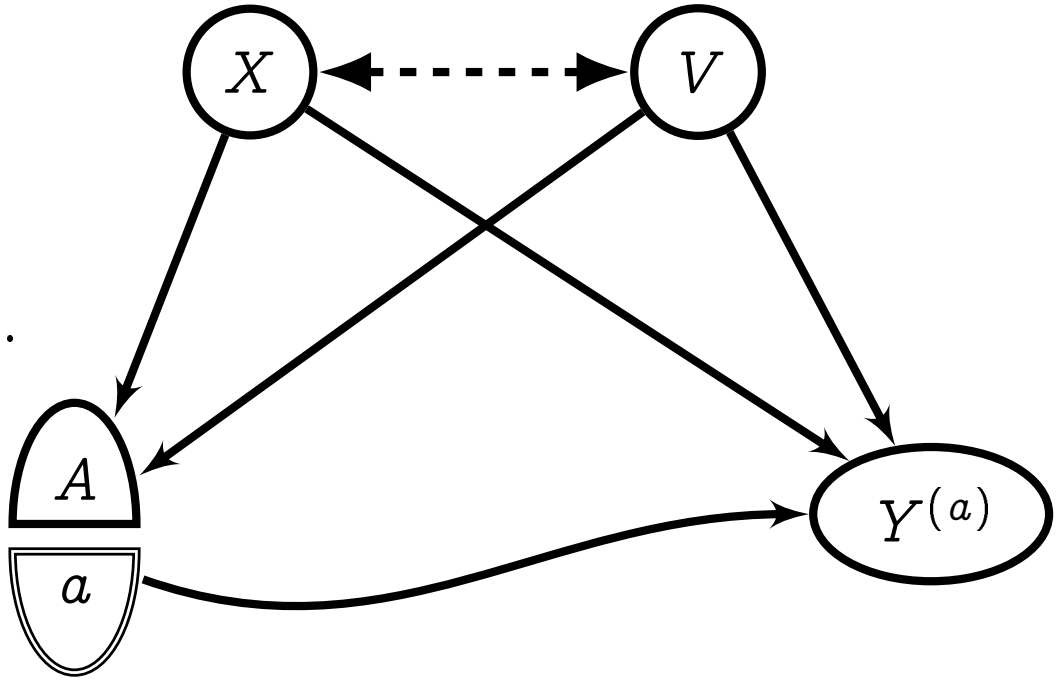
Conditional average treatment effect

Well-specified setting:

$$\mathbb{E}[Y|a,x,v] =: \gamma_0(a,x,v)$$
$$= \langle \gamma_0, \varphi(a) \otimes \varphi(x) \otimes \varphi(v) \rangle .$$

Conditional ATE

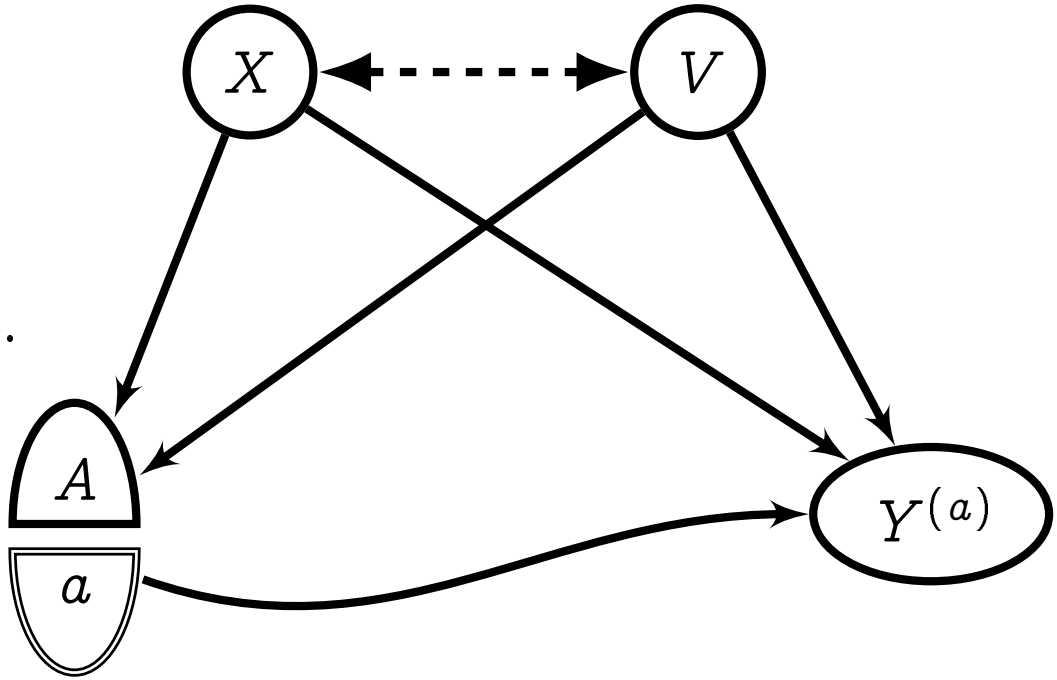
$$\mathrm{CATE}(a,v)$$
$$= \mathbb{E}\left[Y^{(a)} | V = v \right]$$



Conditional average treatment effect

Well-specified setting:

$$\mathbb{E}[Y|a,x,v] =: \gamma_0(a,x,v)$$
$$= \langle \gamma_0, \varphi(a) \otimes \varphi(x) \otimes \varphi(v) \rangle .$$



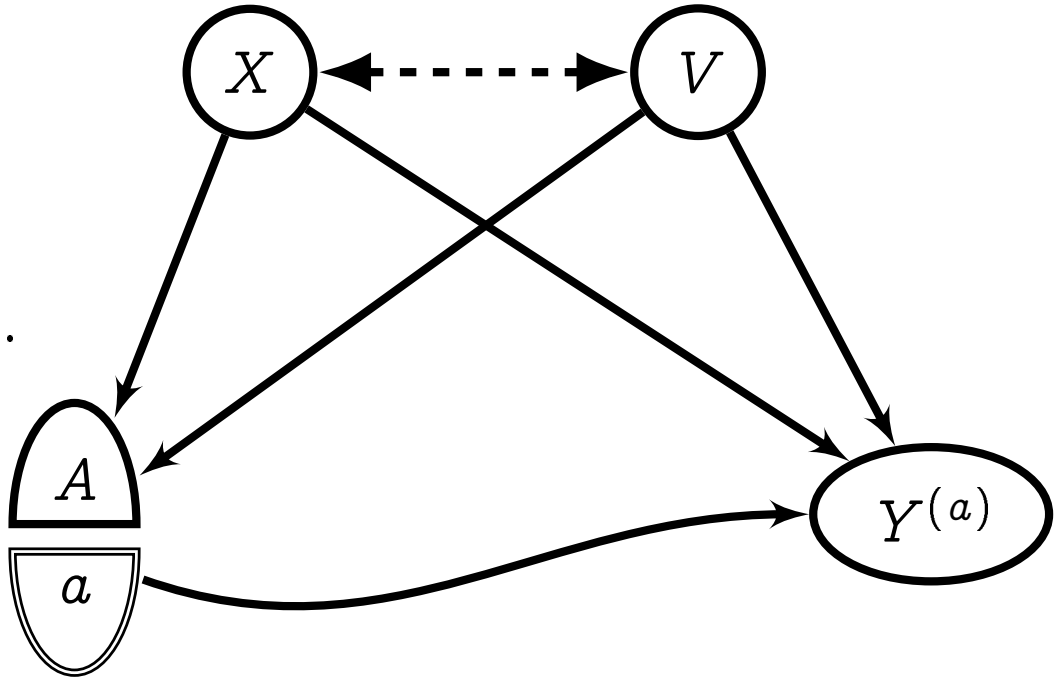
Conditional ATE

$$\text{CATE}(a,v)$$
$$= \mathbb{E}\left[Y^{(a)} | V = v \right]$$
$$= \mathbb{E}\left[\langle \gamma_0, \varphi(a) \otimes \varphi(X) \otimes \varphi(V) \rangle \,|\, V = v \right]$$

Conditional average treatment effect

Well-specified setting:

$$\mathbb{E}[Y \mid a, x, v] =: \gamma_0(a, x, v)$$
$$= \langle \gamma_0, \varphi(a) \otimes \varphi(x) \otimes \varphi(v) \rangle .$$



Conditional ATE

$$\text{CATE}(a, v)$$
$$= \mathbb{E}\left[Y^{(a)} \mid V = v \right]$$
$$= \mathbb{E}\left[\langle \gamma_0, \varphi(a) \otimes \varphi(X) \otimes \varphi(V) \rangle \mid V = v \right]$$
$$= \ldots ?$$

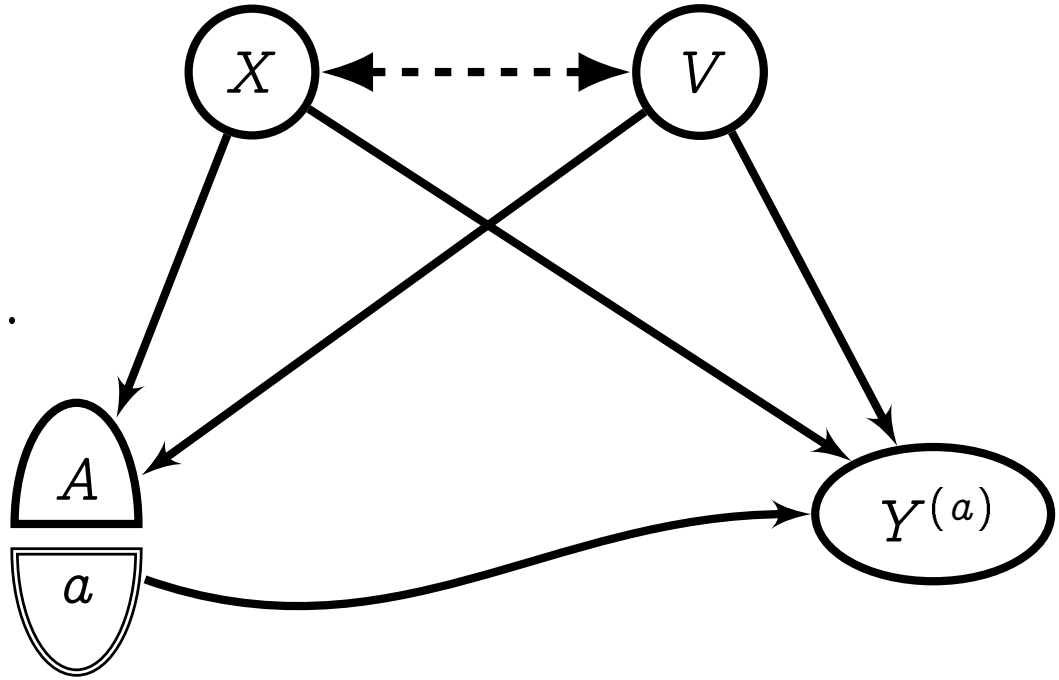
How to take conditional expectation?

Density estimation for $p(X \mid V = v)$? Sample from $p(X \mid V = v)$?

Conditional average treatment effect

Well-specified setting:

$$\mathbb{E}[Y|a,x,v] =: \gamma_0(a,x,v)$$
$$= \langle \gamma_0, \varphi(a) \otimes \varphi(x) \otimes \varphi(v) \rangle .$$



Conditional ATE

$$\text{CATE}(a,v)$$
$$= \mathbb{E}\left[Y^{(a)} | V = v \right]$$
$$= \mathbb{E}\left[\langle \gamma_0, \varphi(a) \otimes \varphi(X) \otimes \varphi(V) \rangle \,|\, V = v \right]$$
$$= \langle \gamma_0, \varphi(a) \otimes \underbrace{\mathbb{E}[\varphi(X)|V=v]}_{\mu_{X|V=v}} \otimes \varphi(v) \rangle$$

Learn conditional mean embedding: $\mu_{X|V=v} := \mathbb{E}_X[\varphi(X)|V=v]$

Regressing *from* feature space *to* feature space

Our goal: an operator $F_0 : \mathcal{H}_\mathcal{V} \to \mathcal{H}_\mathcal{X}$ such that

$$F_0 \varphi(v) = \mu_{X|V=v}$$

Song, Huang, Smola, Fukumizu (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems.

Grunewalder, Lever, Baldassarre, Patterson, G, Pontil (2012). Conditional mean embeddings as regressors.

Grunewalder, G, Shawe-Taylor (2013) Smooth operators.

Li, Meunier, Mollenhauer, G (2022), Optimal Rates for Regularized Conditional Mean Embedding Learning

Regressing *from* feature space *to* feature space

Our goal: an operator $F_0 : \mathcal{H}_\mathcal{V} \to \mathcal{H}_\mathcal{X}$ such that

$$F_0 \varphi(v) = \mu_{X|V=v}$$

Assume

$$F_0 \in \overline{\text{span}\{\varphi(x) \otimes \varphi(v)\}} \iff F_0 \in \text{HS}(\mathcal{H}_\mathcal{V}, \mathcal{H}_\mathcal{X})$$

Implied smoothness assumption:

$$\mathbb{E}[h(X)|V = v] \in \mathcal{H}_\mathcal{V} \quad \forall h \in \mathcal{H}_\mathcal{X}$$

Song, Huang, Smola, Fukumizu (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems.

Grunewalder, Lever, Baldassarre, Patterson, G, Pontil (2012). Conditional mean embeddings as regressors.

Grunewalder, G, Shawe-Taylor (2013) Smooth operators.

Li, Meunier, Mollenhauer, G (2022), Optimal Rates for Regularized Conditional Mean Embedding Learning

Regressing *from* feature space *to* feature space

Our goal: an operator $F_0 : \mathcal{H}_\mathcal{V} \to \mathcal{H}_\mathcal{X}$ such that

$$F_0 \varphi(v) = \mu_{X|V=v}$$

Assume

$$F_0 \in \overline{\text{span}\{\varphi(x) \otimes \varphi(v)\}} \iff F_0 \in \text{HS}(\mathcal{H}_\mathcal{V}, \mathcal{H}_\mathcal{X})$$

Implied smoothness assumption:

$$\mathbb{E}[h(X)|V = v] \in \mathcal{H}_\mathcal{V} \quad \forall h \in \mathcal{H}_\mathcal{X}$$

A Smooth Operator

Song, Huang, Smola, Fukumizu (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems.

Grunewalder, Lever, Baldassarre, Patterson, G, Pontil (2012). Conditional mean embeddings as regressors.

Grunewalder, G, Shawe-Taylor (2013) Smooth operators.

Li, Meunier, Mollenhauer, G (2022), Optimal Rates for Regularized Conditional Mean Embedding Learning

Regressing *from* feature space *to* feature space

Our goal: an operator $F_0 : \mathcal{H}_\mathcal{V} \to \mathcal{H}_\mathcal{X}$ such that

$$F_0 \varphi(v) = \mu_{X|V=v}$$

Assume

$$F_0 \in \overline{\text{span}\{\varphi(x) \otimes \varphi(v)\}} \iff F_0 \in \text{HS}(\mathcal{H}_\mathcal{V}, \mathcal{H}_\mathcal{X})$$

Implied smoothness assumption:

$$\mathbb{E}[h(X)|V = v] \in \mathcal{H}_\mathcal{V} \quad \forall h \in \mathcal{H}_\mathcal{X}$$

Kernel ridge regression from $\varphi(v)$ to *infinite* features $\varphi(x)$:

$$\widehat{F} = \underset{F \in HS}{\text{argmin}} \sum_{\ell=1}^{n} \|\varphi(x_\ell) - F\varphi(v_\ell)\|^2_{\mathcal{H}_\mathcal{X}} + \lambda_2 \|F\|^2_{HS}$$

Song, Huang, Smola, Fukumizu (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems.

Grunewalder, Lever, Baldassarre, Patterson, G, Pontil (2012). Conditional mean embeddings as regressors.

Grunewalder, G, Shawe-Taylor (2013) Smooth operators.

Li, Meunier, Mollenhauer, G (2022), Optimal Rates for Regularized Conditional Mean Embedding Learning

Regressing *from* feature space *to* feature space

Our goal: an operator $F_0 : \mathcal{H}_\mathcal{V} \to \mathcal{H}_\mathcal{X}$ such that

$$F_0 \varphi(v) = \mu_{X|V=v}$$

Assume

$$F_0 \in \overline{\text{span}\{\varphi(x) \otimes \varphi(v)\}} \iff F_0 \in \text{HS}(\mathcal{H}_\mathcal{V}, \mathcal{H}_\mathcal{X})$$

Implied smoothness assumption:

$$\mathbb{E}[h(X)|V = v] \in \mathcal{H}_\mathcal{V} \quad \forall h \in \mathcal{H}_\mathcal{X}$$

Kernel ridge regression from $\varphi(v)$ to *infinite* features $\varphi(x)$:

$$\widehat{F} = \underset{F \in HS}{\text{argmin}} \sum_{\ell=1}^{n} \|\varphi(x_\ell) - F\varphi(v_\ell)\|^2_{\mathcal{H}_\mathcal{X}} + \lambda_2 \|F\|^2_{HS}$$

Ridge regression solution:

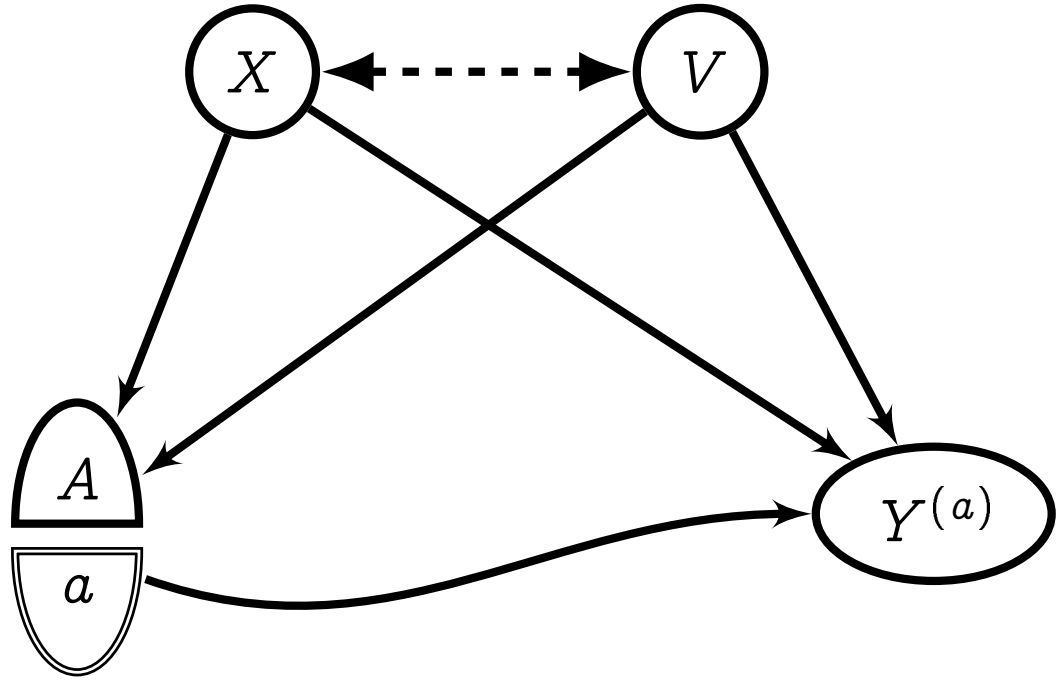
$$\mu_{X|V=v} := \mathbb{E}[\varphi(X)|V = v] \approx \widehat{F}\varphi(v) = \sum_{\ell=1}^{n} \varphi(x_\ell)\beta_\ell(v)$$

$$\beta(v) = [K_{VV} + \lambda_2 I]^{-1} k_{Vv}$$

Conditional ATE: example

US job corps:

- X: confounder/context (education, marital status, ...)
- A: treatment (training hours)
- Y: outcome (percent employed)
- V: age

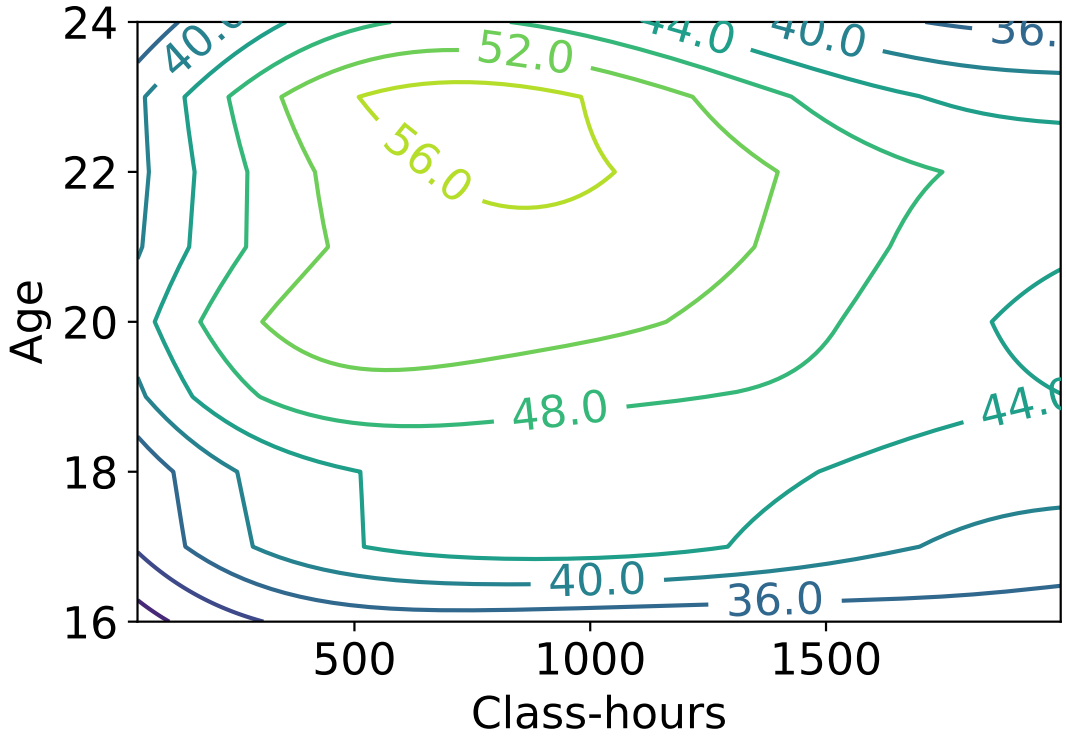


Empirical CATE:

$$\widehat{\mathrm{CATE}}(a, v) = \langle \hat{\gamma}_0, \varphi(a) \otimes \underbrace{\widehat{F}\varphi(v)}_{\widehat{\mathbb{E}}[\varphi(X) \mid V = v]} \otimes \varphi(v) \rangle$$

(with consistency guarantees: see paper!)

Conditional ATE: results



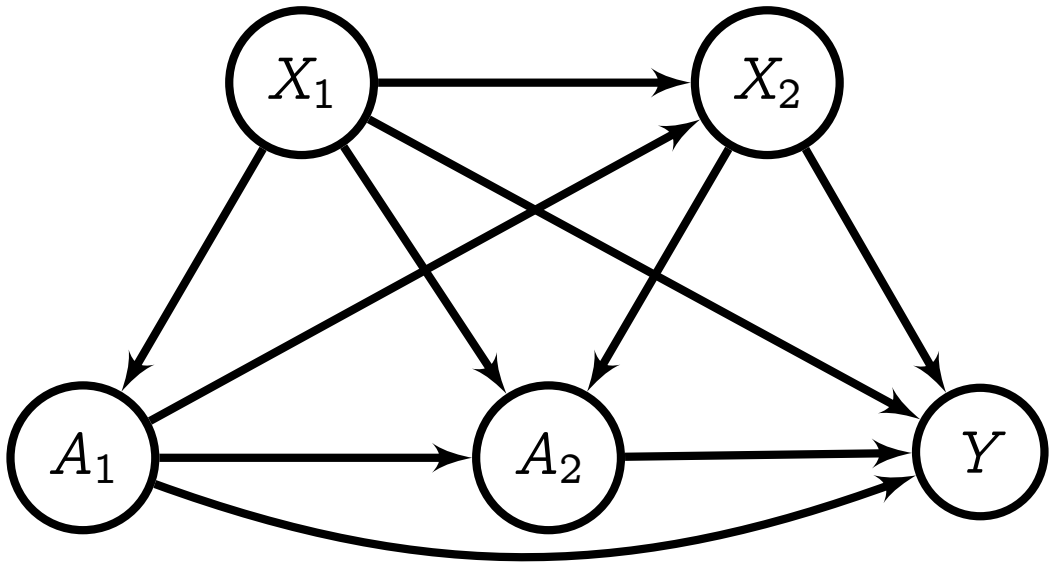
Average percentage employment $Y^{(a)}$ for class hours a, conditioned on age v. Given around 12-14 weeks of classes:

- 16 y/o: employment increases from 28% to at most 36%.
- 22 y/o: percent employment increases from 40% to 56%.

Singh, Xu, G (2023)

...dynamic treatment effect...

Dynamic treatment effect: sequence A_1, A_2 of treatments.



- potential outcomes $Y^{(a_1)}$, $Y^{(a_2)}$, $Y^{(a_1,a_2)}$,
- counterfactuals $\mathbb{E}\left[Y^{(a_1',a_2')} | A_1 = a_1, A_2 = a_2\right]$...

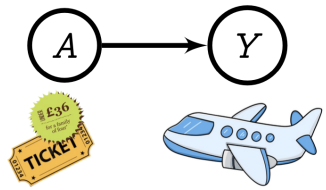
(c.f. the Robins G-formula)

Singh, Xu, G. (Bernoulli 2025) Kernel Methods for Multistage Causal Inference: Mediation Analysis and Dynamic Treatment Effects

What if there are hidden confounders?

Illustration: ticket prices for air travel

Ticket price A, seats sold Y.

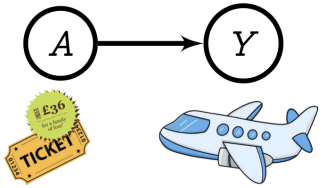


What is the effect on seats sold $Y^{(a)}$ of intervening on price a?

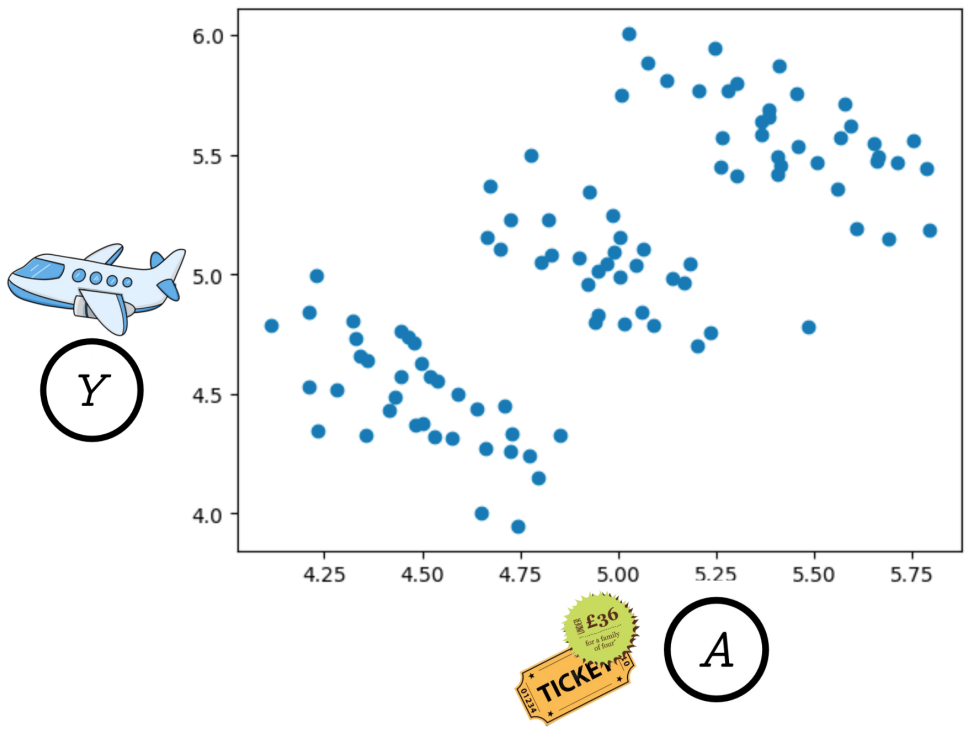
Simplification of example from Hartford, Lewis, Leyton-Brown, Taddy (2017): Deep IV: A Flexible Approach for Counterfactual Prediction.

Illustration: ticket prices for air travel

Ticket price A, seats sold Y.



What is the effect on seats sold $Y^{(a)}$ of intervening on price a?

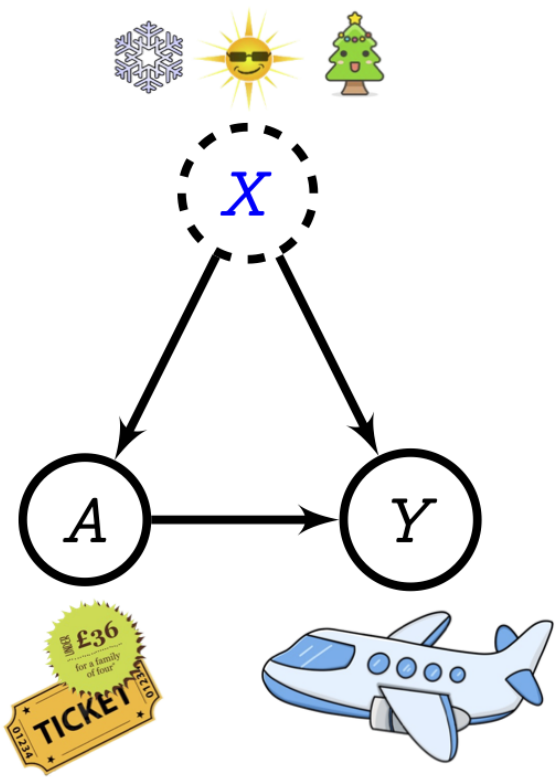


Simplification of example from Hartford, Lewis, Leyton-Brown, Taddy (2017): Deep IV: A Flexible Approach for Counterfactual Prediction.

Illustration: ticket prices for air travel

Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.

- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$



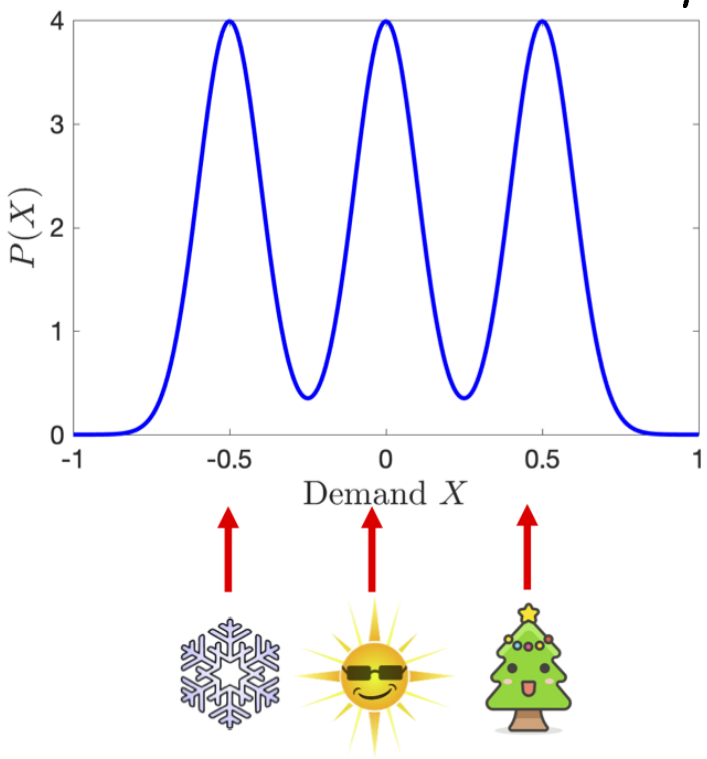
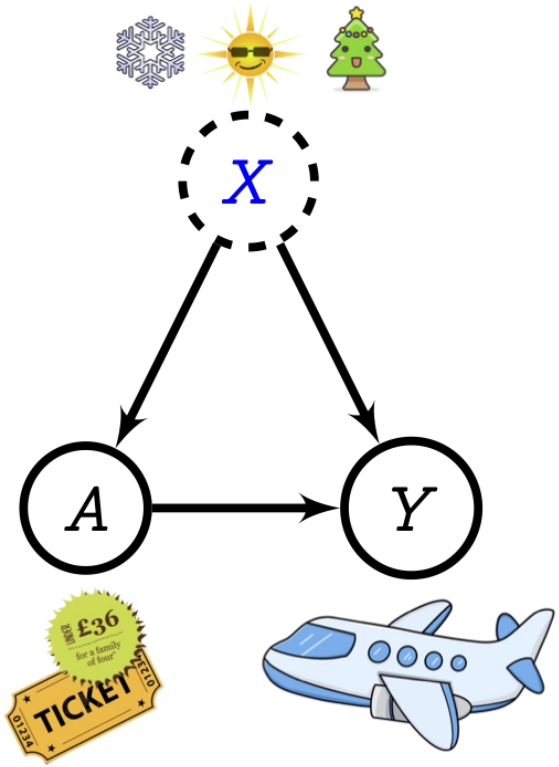


Illustration: ticket prices for air travel

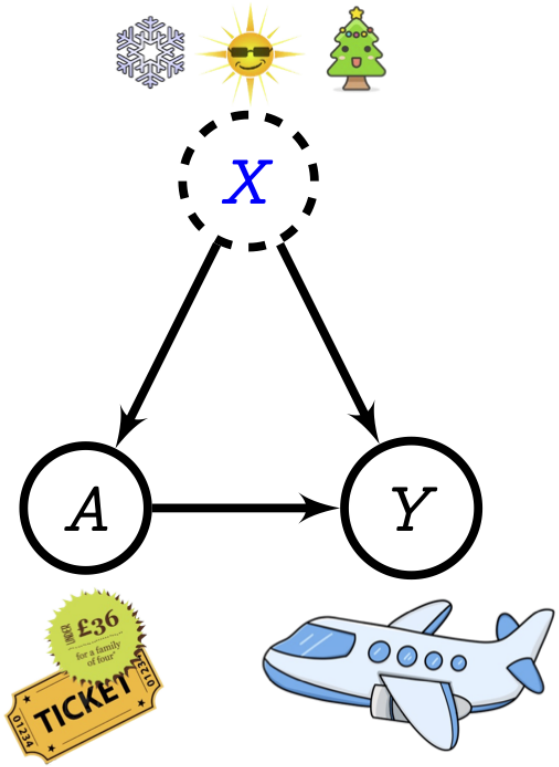
Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$
- Price:
  $A = X + Z,$
  $Z \sim \mathcal{N}(5, 0.04)$

Illustration: ticket prices for air travel

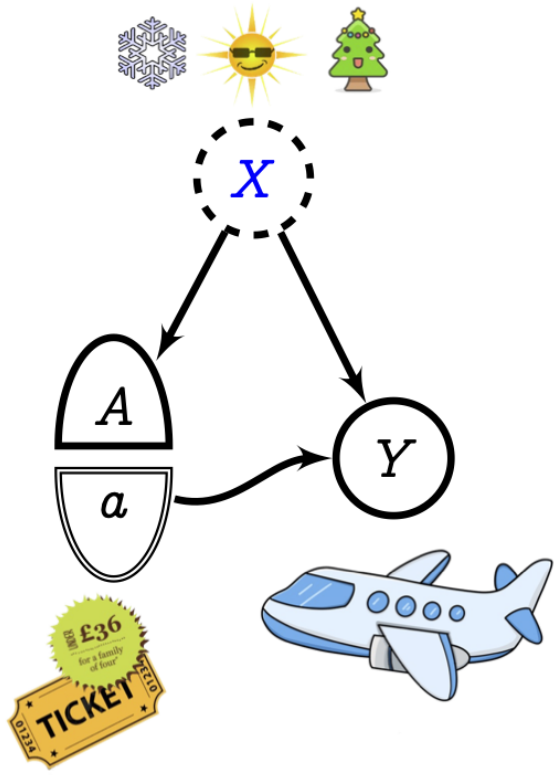
Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$
- Price:
  $A = X + Z,$
  $Z \sim \mathcal{N}(5, 0.04)$
- Seats sold:
  $Y = 10 - A + 2X$

Illustration: ticket prices for air travel

Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



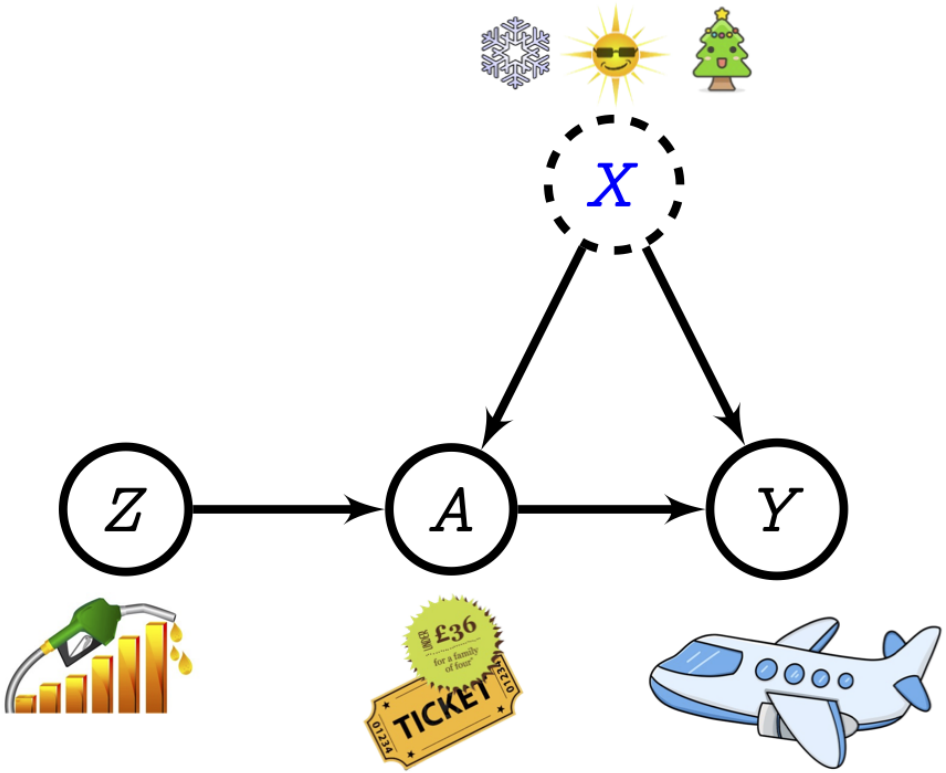
- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$
- Price:
  $A = X + Z,$
  $Z \sim \mathcal{N}(5, 0.04)$
- Seats sold:
  $Y = 10 - A + 2X$

Average treatment effect:

$$\mathrm{ATE}(a) = \mathbb{E}[Y^{(a)}] = \int (10 - a + 2X)\, dp(X) = 10 - a$$

Illustration: ticket prices for air travel

Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



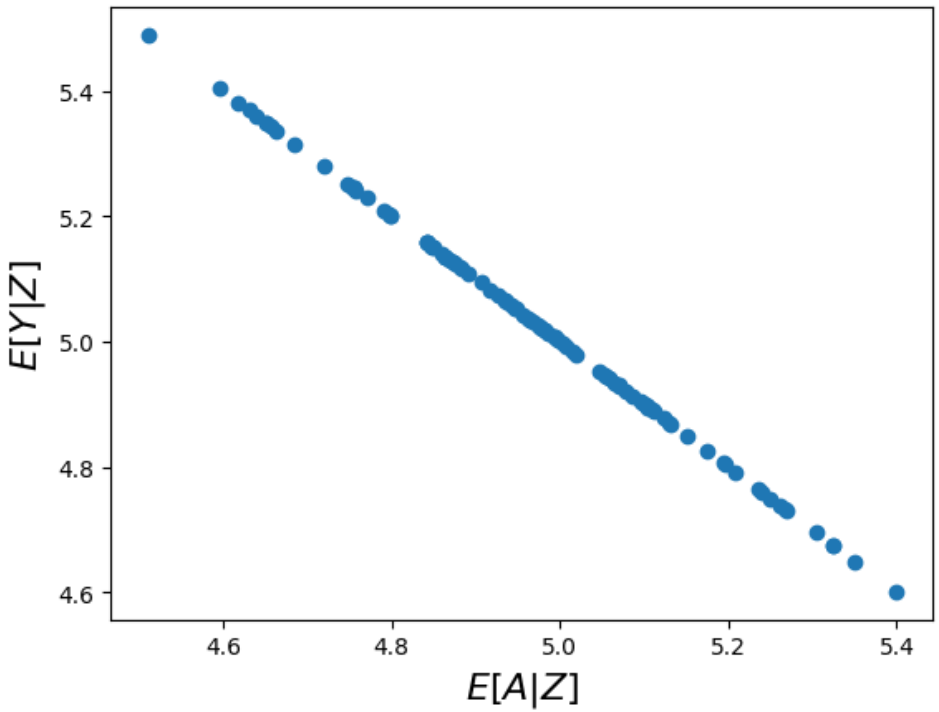
- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$
- Price:
  $A = X + Z,$
  $Z \sim \mathcal{N}(5, 0.04)$
- Seats sold:
  $Y = 10 - A + 2X$

Z is an instrument (cost of fuel). Condition on Z,

$$\mathbb{E}[Y|Z] = 10 - \mathbb{E}[A|Z] + 2\underbrace{\mathbb{E}[X|Z]}_{=0}$$

Illustration: ticket prices for air travel

Unobserved variable X =desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



- Desire for travel:
  $X \sim \mathcal{N}(\mu, 0.01)$
  $\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$
- Price:
  $A = X + Z,$
  $Z \sim \mathcal{N}(5, 0.04)$
- Seats sold:
  $Y = 10 - A + 2X$

Z is an instrument (cost of fuel). Condition on Z,

$$\mathbb{E}[Y|Z] = 10 - \mathbb{E}[A|Z] + 2\underbrace{\mathbb{E}[X|Z]}_{=0}$$

Regressing from $\mathbb{E}[A|Z]$ to $\mathbb{E}[Y|Z]$ recovers causal relation!

Instrumental variable regression

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2021

© Nobel Prize Outreach. Photo: Paul Kennedy

David Card

Prize share: 1/2

© Nobel Prize Outreach. Photo: Risdon Photography

Joshua D. Angrist

Prize share: 1/4

© Nobel Prize Outreach. Photo: Paul Kennedy

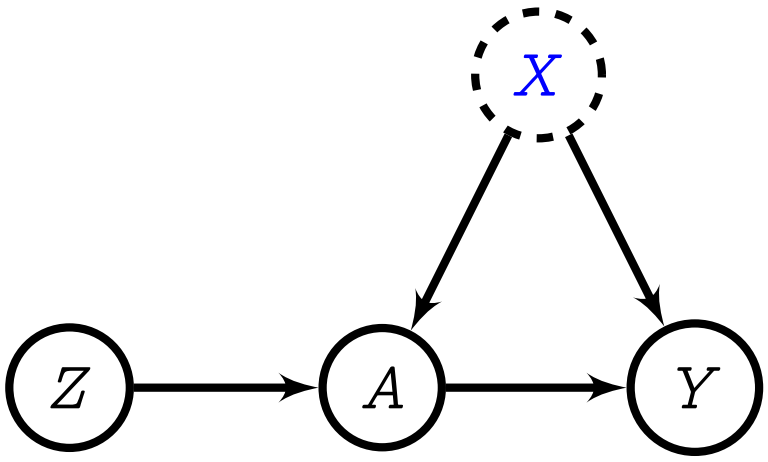
Guido W. Imbens

Prize share: 1/4

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2021 was divided, one half awarded to David Card "for his empirical contributions to labour economics", the other half jointly to Joshua D. Angrist and Guido W. Imbens "for their methodological contributions to the analysis of causal relationships"

Instrumental variable regression with NN features

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



Assumptions

$$\mathbb{E}[X|Z] = 0$$

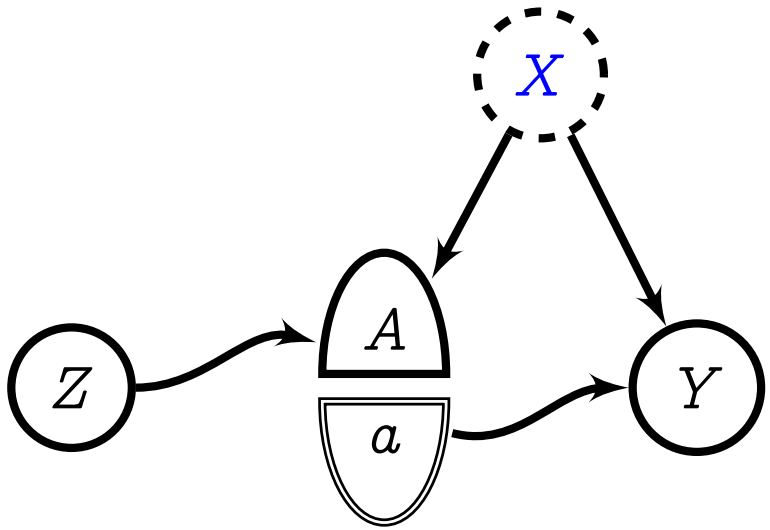
$$Z \not\perp\!\!\!\perp A$$

$$(Y \perp\!\!\!\perp Z|A)_{G_{\bar{A}}}$$

$$Y = \gamma^\top \phi_\theta(A) + X$$

Instrumental variable regression with NN features

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



Assumptions

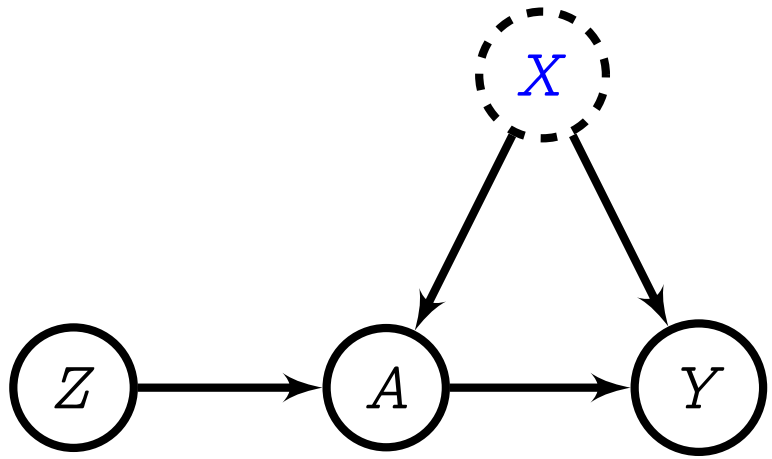
$$\mathbb{E}[X|Z] = 0$$
$$Z \not\perp\!\!\!\perp A$$
$$(Y \perp\!\!\!\perp Z|A)_{G_{\bar{A}}}$$
$$Y = \gamma^\top \phi_\theta(A) + X$$

Average causal effect:

$$\mathrm{ATE}(a) = \int \mathbb{E}(Y|X,a)\,dp(X) = \gamma^\top \phi_\theta(a)$$

Instrumental variable regression with NN features

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



Assumptions

$$\mathbb{E}[X|Z] = 0$$

$$Z \not\perp\!\!\!\perp A$$

$$(Y \perp\!\!\!\perp Z|A)_{G_{\bar{A}}}$$

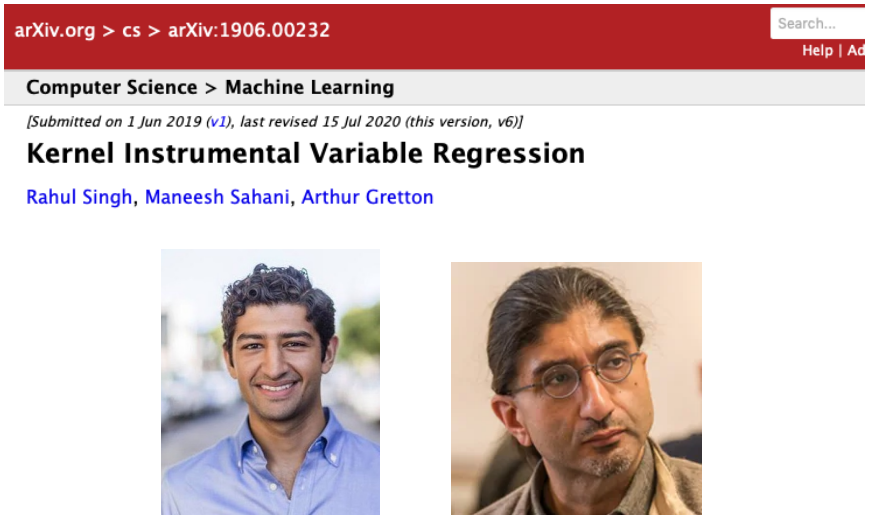
$$Y = \gamma^\top \phi_\theta(A) + X$$

IV regression: Condition both sides on Z,

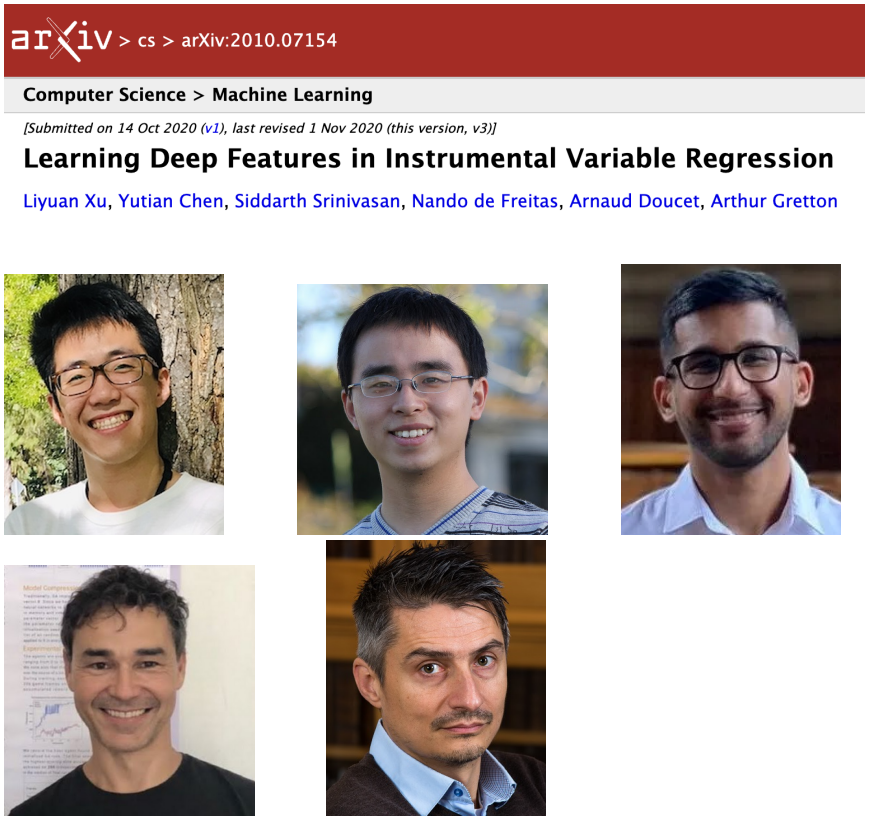
$$\mathbb{E}[Y|Z] = \gamma^\top \mathbb{E}[\phi_\theta(A)|Z] + \underbrace{\mathbb{E}[X|Z]}_{=0}$$

Two-stage least squares for IV regression

Kernel features (NeurIPS 2019):



NN features (ICLR 2021):

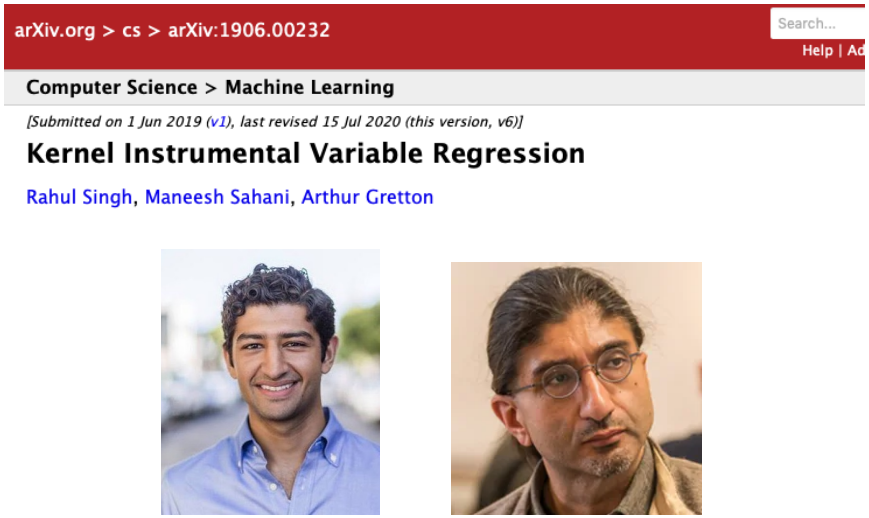


Code for NN and kernel IV methods:

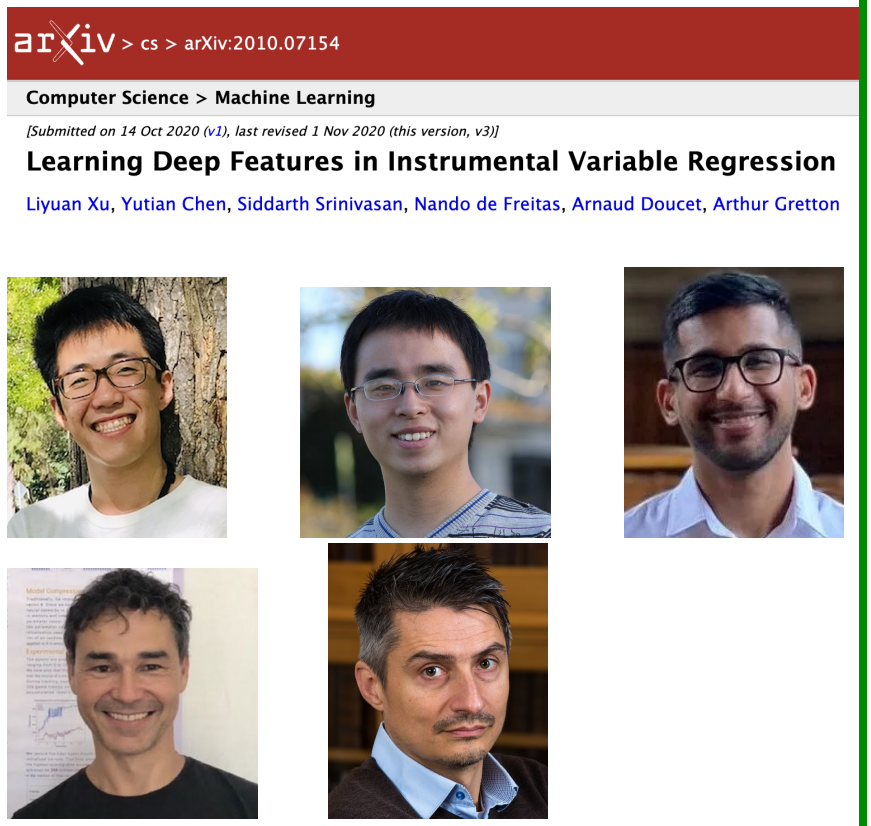
https://github.com/liyuan9988/DeepFeatureIV/

Two-stage least squares for IV regression

Kernel features (NeurIPS 2019):



NN features (ICLR 2021):



Code for NN and kernel IV methods:

https://github.com/liyuan9988/DeepFeatureIV/

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[\left(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z]\right)^2\right] + \lambda_2\|\gamma\|^2$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Challenge: how to learn θ?

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Challenge: how to learn θ?

From Stage 2 regression?

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Challenge: how to learn θ?

From Stage 2 regression?

...which requires $\mathbb{E}[\phi_\theta(A)|Z]$ from Stage 1 regression

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Challenge: how to learn θ?

From Stage 2 regression?
...which requires $\mathbb{E}[\phi_\theta(A)|Z]$ from Stage 1 regression
...which requires $\phi_\theta(A)$... which requires θ...

IV using neural net features

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

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

Challenge: how to learn θ?

From Stage 2 regression?
...which requires $\mathbb{E}[\phi_\theta(A)|Z]$ from Stage 1 regression
...which requires $\phi_\theta(A)$... which requires θ...

Use the linear final layers! (i.e. γ and F)

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

IV using neural net features

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E}\left[\|\phi_\theta(A) - F\phi_\zeta(Z)\|^2\right] + \lambda_1\|F\|_{HS}^2$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

IV using neural net features

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E}\left[\|\phi_\theta(A) - F\phi_\zeta(Z)\|^2\right] + \lambda_1\|F\|_{HS}^2$$

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

$$\hat{F}_{\theta,\zeta} = C_{AZ}(C_{ZZ} + \lambda_1 I)^{-1} \qquad C_{AZ} = \mathbb{E}[\phi_\theta(A)\phi_\zeta^\top(Z)]$$
$$C_{ZZ} = \mathbb{E}[\phi_\zeta(Z)\phi_\zeta^\top(Z)]$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

IV using neural net features

Stage 1 regression: learn NN features $\phi_\zeta(Z)$ and linear layer F:

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E}\left[\|\phi_\theta(A) - F\phi_\zeta(Z)\|^2\right] + \lambda_1\|F\|_{HS}^2$$

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

$$\hat{F}_{\theta,\zeta} = C_{AZ}(C_{ZZ} + \lambda_1 I)^{-1} \qquad C_{AZ} = \mathbb{E}[\phi_\theta(A)\phi_\zeta^\top(Z)]$$
$$C_{ZZ} = \mathbb{E}[\phi_\zeta(Z)\phi_\zeta^\top(Z)]$$

Plug $\hat{F}_{\theta,\zeta}$ into S1 loss, take gradient steps for ζ (...but not θ...)

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021) Learning Deep Features in Instrumental Variable Regresion

Stage 2: IV regression

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathcal{L}_2(\gamma, \theta) = \mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Alternative: functional bilevel optimization: Petrulionyte, Mairal, Arbel (2024)

Stage 2: IV regression

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\mathcal{L}_2(\gamma, \theta) = \mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2$$
$$= \mathbb{E}_{YZ}[(Y - \gamma^\top \underbrace{\hat{F}_{\theta,\zeta}\phi_\zeta(Z)}_{\text{Stage 1}})^2] + \lambda_2\|\gamma\|^2$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Alternative: functional bilevel optimization: Petrulionyte, Mairal, Arbel (2024)

Stage 2: IV regression

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2 \\ &= \mathbb{E}_{YZ}[(Y - \gamma^\top \hat{F}_{\theta,\zeta}\phi_\zeta(Z))^2] + \lambda_2\|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$ in closed form wrt ϕ_θ:

$$\hat{\gamma}_\theta := \widetilde{C}_{YA|Z}(\widetilde{C}_{AA|Z} + \lambda_2 I)^{-1} \qquad \widetilde{C}_{YA|Z} = \mathbb{E}\left[Y\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

$$\widetilde{C}_{AA|Z} = \mathbb{E}\left[[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Alternative: functional bilevel optimization: Petrulionyte, Mairal, Arbel (2024)

Stage 2: IV regression

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2 \\ &= \mathbb{E}_{YZ}[(Y - \gamma^\top \hat{F}_{\theta,\zeta}\phi_\zeta(Z))^2] + \lambda_2\|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$ in closed form wrt ϕ_θ:

$$\hat{\gamma}_\theta := \widetilde{C}_{YA|Z}(\widetilde{C}_{AA|Z} + \lambda_2 I)^{-1} \qquad \widetilde{C}_{YA|Z} = \mathbb{E}\left[Y\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

$$\widetilde{C}_{AA|Z} = \mathbb{E}\left[[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

From linear final layers in Stages 1,2:

Learn $\phi_\theta(A)$ by plugging $\hat{\gamma}_\theta$ into S2 loss, taking gradient steps for θ

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Alternative: functional bilevel optimization: Petrulionyte, Mairal, Arbel (2024)

Stage 2: IV regression

Stage 2 regression (IV): learn NN features $\phi_\theta(A)$ and linear layer γ to obtain Y with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ}\left[(Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2\right] + \lambda_2\|\gamma\|^2 \\ &= \mathbb{E}_{YZ}[(Y - \gamma^\top \hat{F}_{\theta,\zeta}\phi_\zeta(Z))^2] + \lambda_2\|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$ in closed form wrt ϕ_θ:

$$\hat{\gamma}_\theta := \widetilde{C}_{YA|Z}(\widetilde{C}_{AA|Z} + \lambda_2 I)^{-1} \qquad \widetilde{C}_{YA|Z} = \mathbb{E}\left[Y\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

$$\widetilde{C}_{AA|Z} = \mathbb{E}\left[[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]\,[\hat{F}_{\theta,\zeta}\varphi_\zeta(Z)]^\top\right]$$

From linear final layers in Stages 1,2:

Learn $\phi_\theta(A)$ by plugging $\hat{\gamma}_\theta$ into S2 loss, taking gradient steps for θ

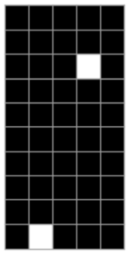
....but ζ changes with θ

...so alternate first and second stages until convergence.

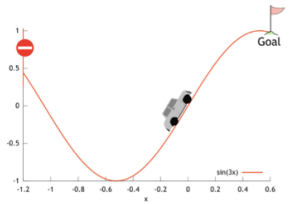
Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Alternative: functional bilevel optimization: Petrulionyte, Mairal, Arbel (2024)

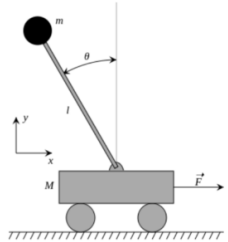
Neural IV in reinforcement learning



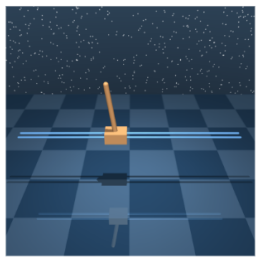
(a) Catch



(b) Mountain Car



(c) Cartpole



(a) Cartpole Swingup

(b) Cheetah Run

(c) Humanoid Run

(d) Walker Walk

Policy evaluation: want Q-value:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \middle| S_0 = s, A_0 = a\right]$$

for policy $\pi(A|S = s)$.

Osband et al (2019). Behaviour suite for reinforcement learning.https://github.com/deepmind/bsuite
Tassa et al. (2020). dm_control:Software and tasks for continuous control.
https://github.com/deepmind/dm_control

Application of IV: reinforcement learning

Q value is a minimizer of Bellman loss

$$\mathcal{L}_{\text{Bellman}} = \mathbb{E}_{SAR}\left[\left(R + \gamma[\mathbb{E}\left[Q^{\pi}(S', A')|S, A\right] - Q^{\pi}(S, A)\right)^2\right].$$

Corresponds to "IV-like" problem

$$\mathcal{L}_{\text{Bellman}} = \mathbb{E}_{YZ}\left[\left(Y - \mathbb{E}[f(X)|Z]\right)^2\right]$$

with

$$\begin{aligned} Y &= R, \\ X &= (S', A', S, A) \\ Z &= (S, A), \\ f_0(X) &= Q^{\pi}(s, a) - \gamma Q^{\pi}(s', a') \end{aligned}$$

RL experiments and data:

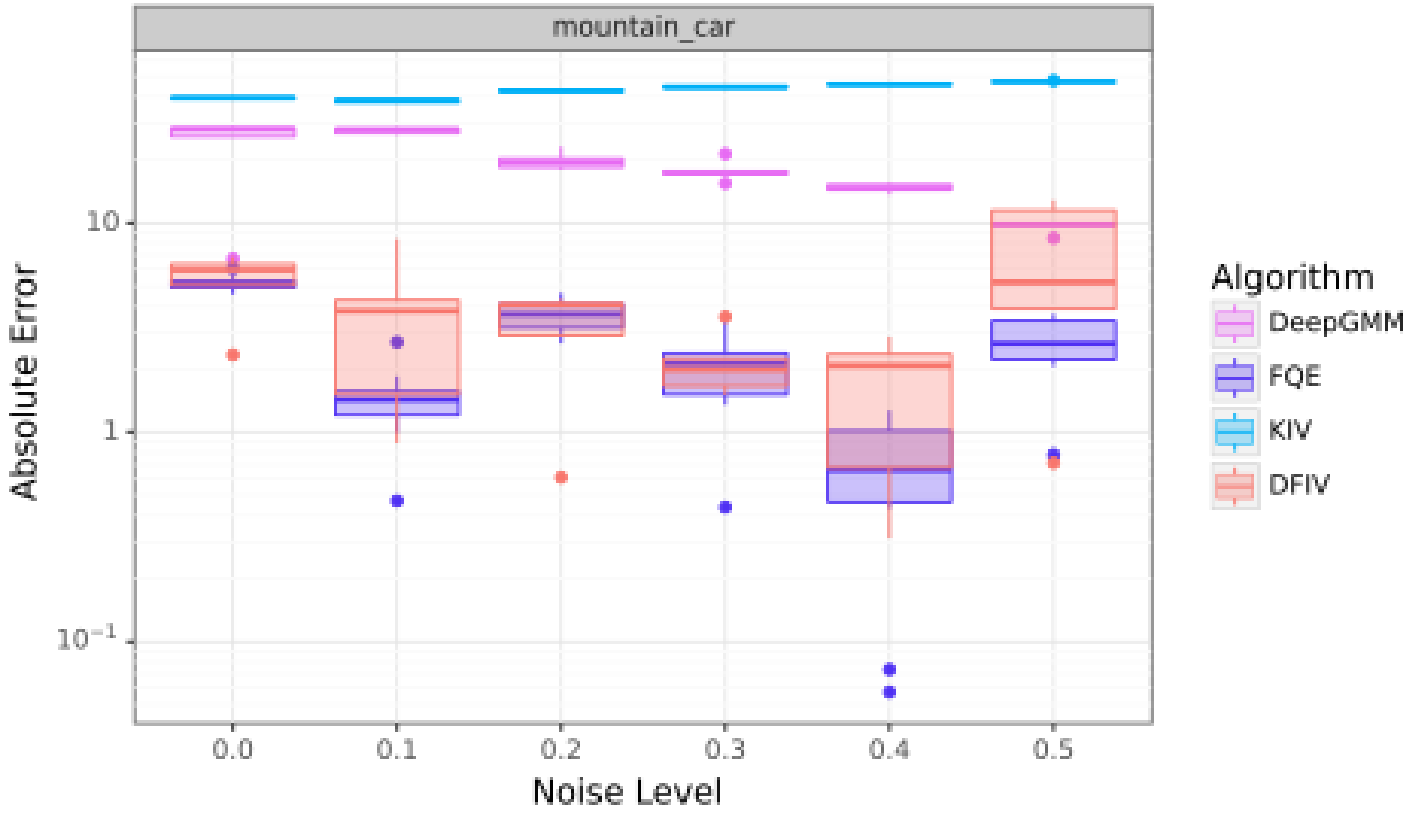
https://github.com/liyuan9988/IVOPEwithACME

Bradtke and Barto (1996). Linear least-squares algorithms for temporal difference learning.

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Chen, Xu, Gulcehre, Le Paine, G, De Freitas, Doucet (2022). On Instrumental Variable Regression for Deep Offline Policy Evaluation.

Results on mountain car problem



Good performance compared with FQE.

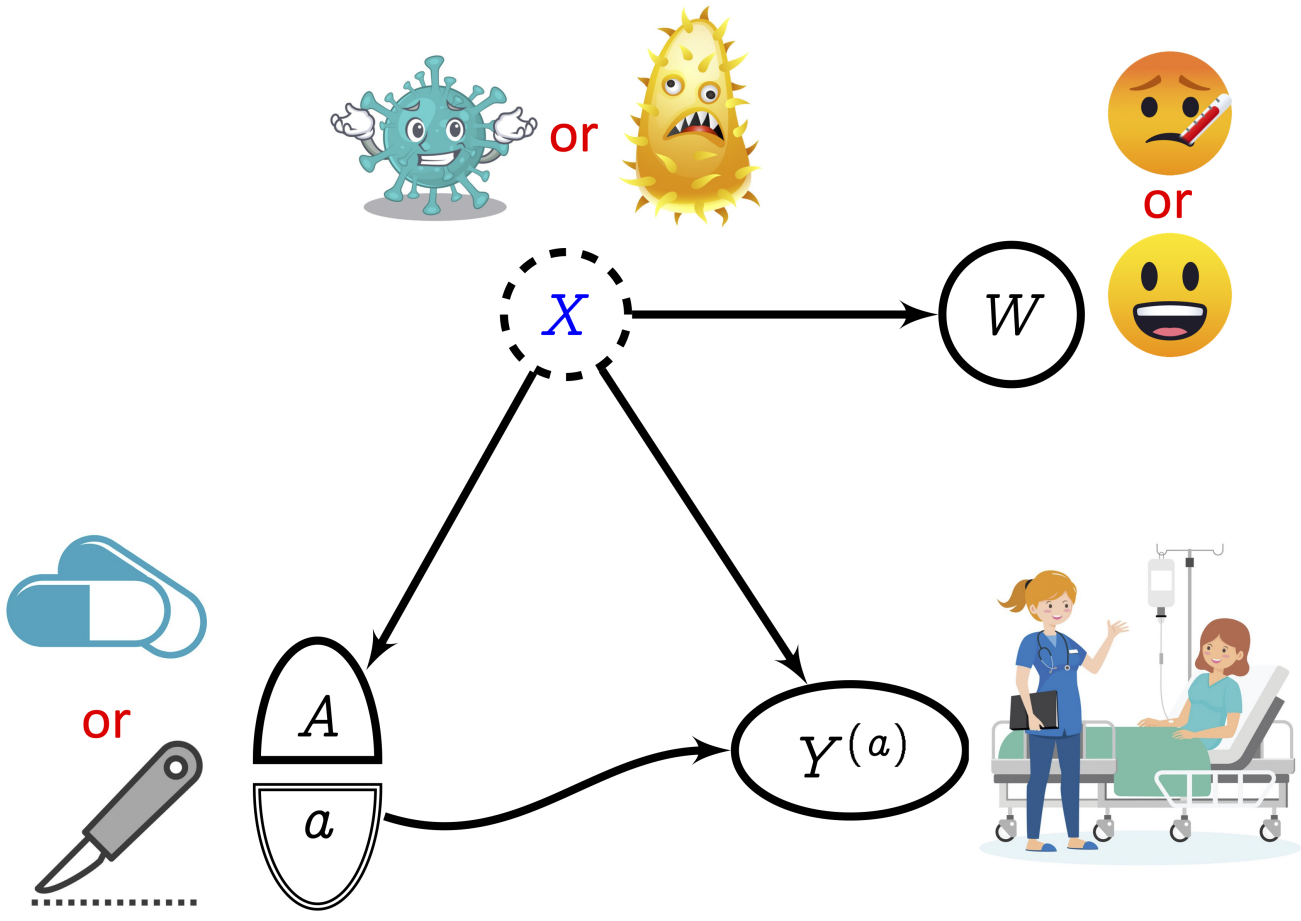
Warning: IV assumption can fail when regression underfits. See papers for details.

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Chen, Xu, Gulcehre, Le Paine, G, De Freitas, Doucet (2022). On Instrumental Variable Regression for Deep Offline Policy Evaluation.

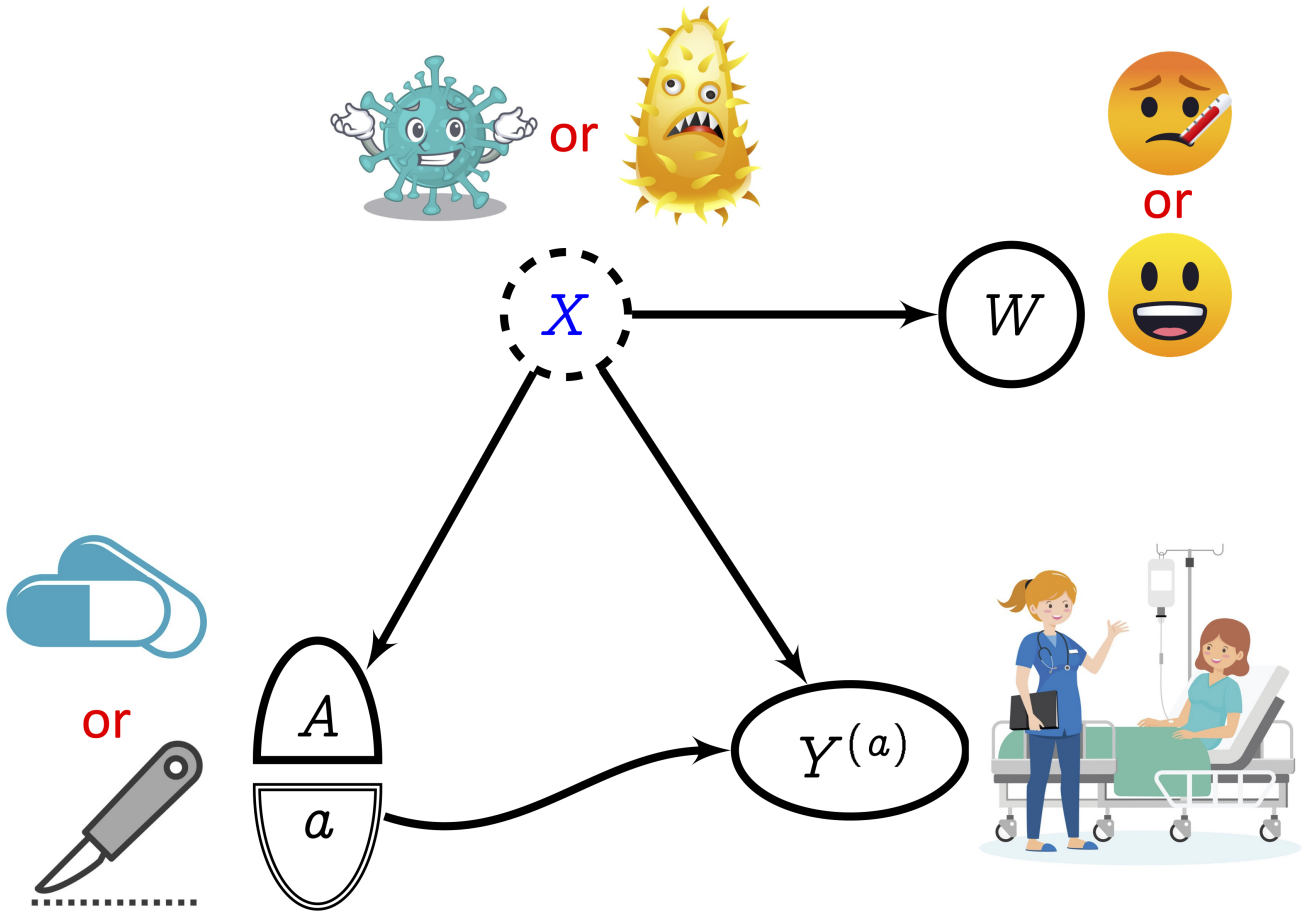
...but seriously, what if there are hidden confounders?

We record symptom W, not disease X



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

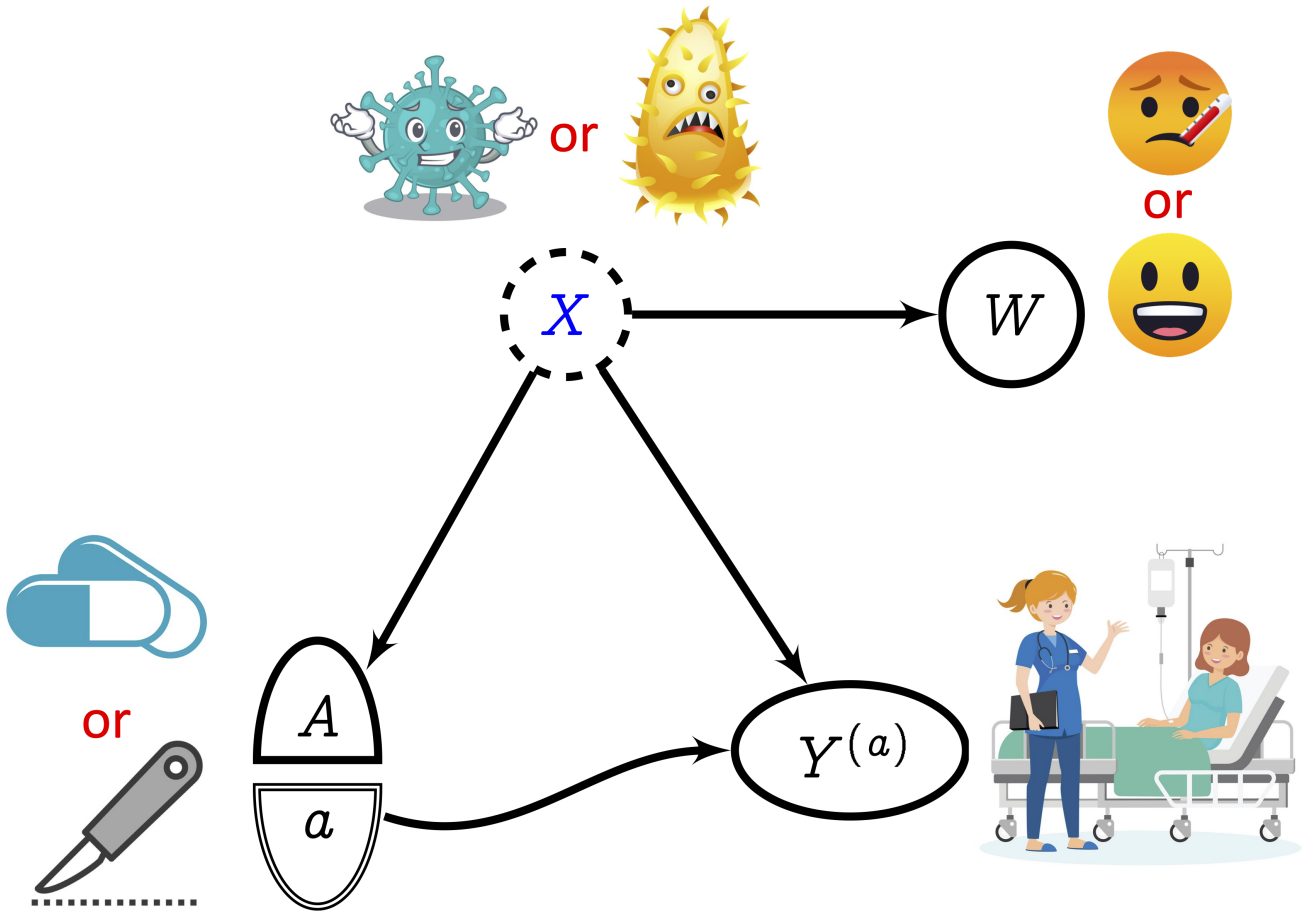
We record symptom W, not disease X



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

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

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

Proxy causal learning (negative controls)

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)

Proxy causal learning (negative controls)

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)

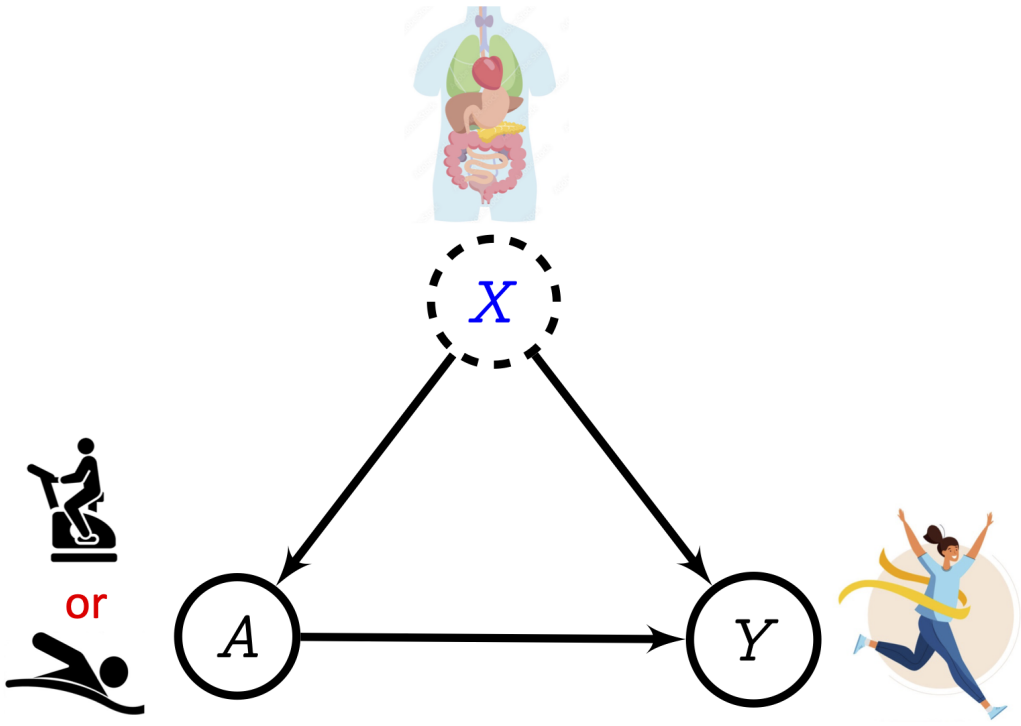
Don't ~~meet your heroes~~ model your hidden variables!

What are proxies, and when are they useful?

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

In this example:

- X: true physical status
- A: exercise regimes
- Y: fitness goal

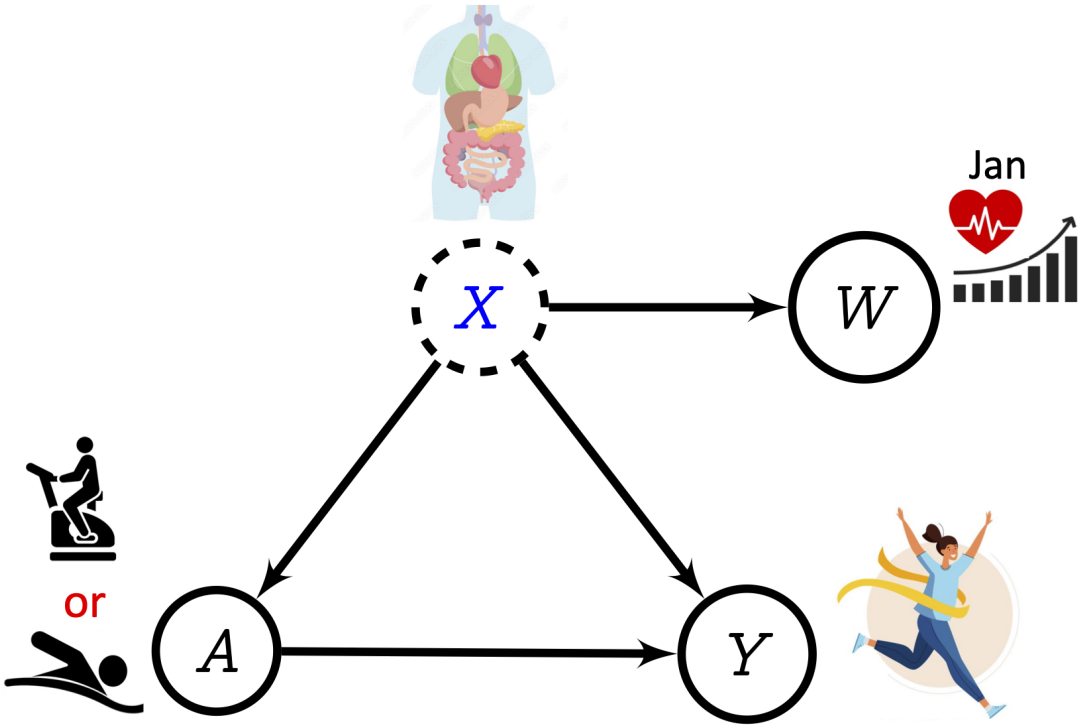


What are proxies, and when are they useful?

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

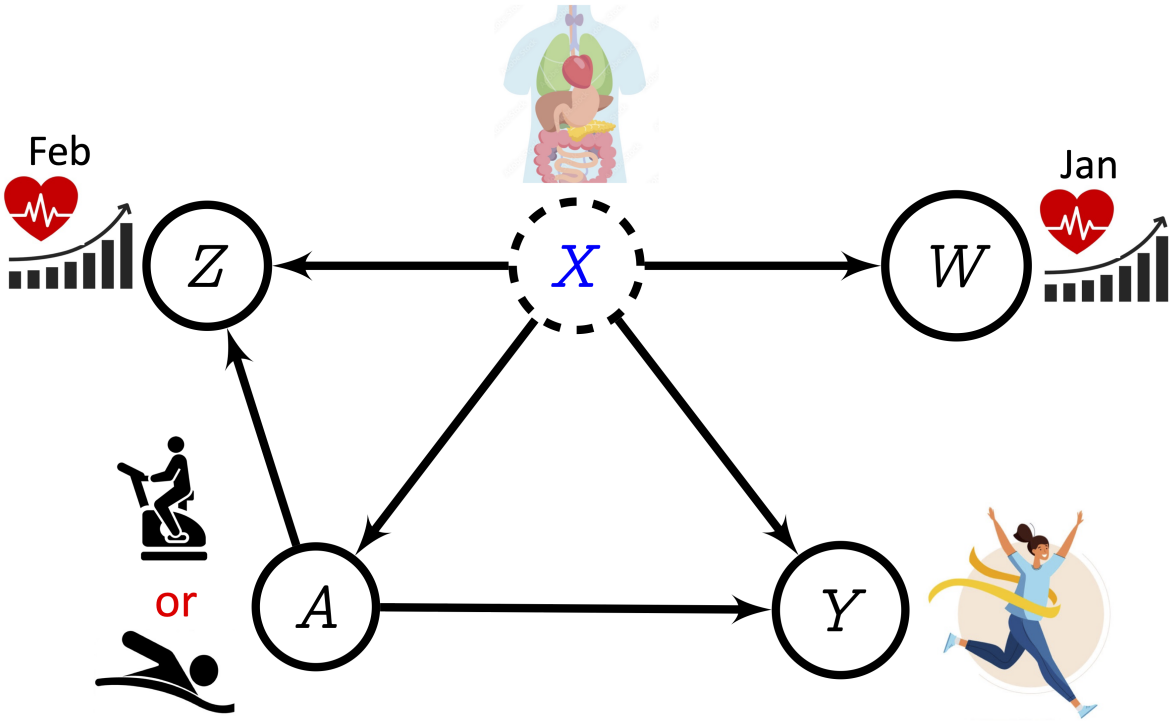


What are proxies, and when are they useful?

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

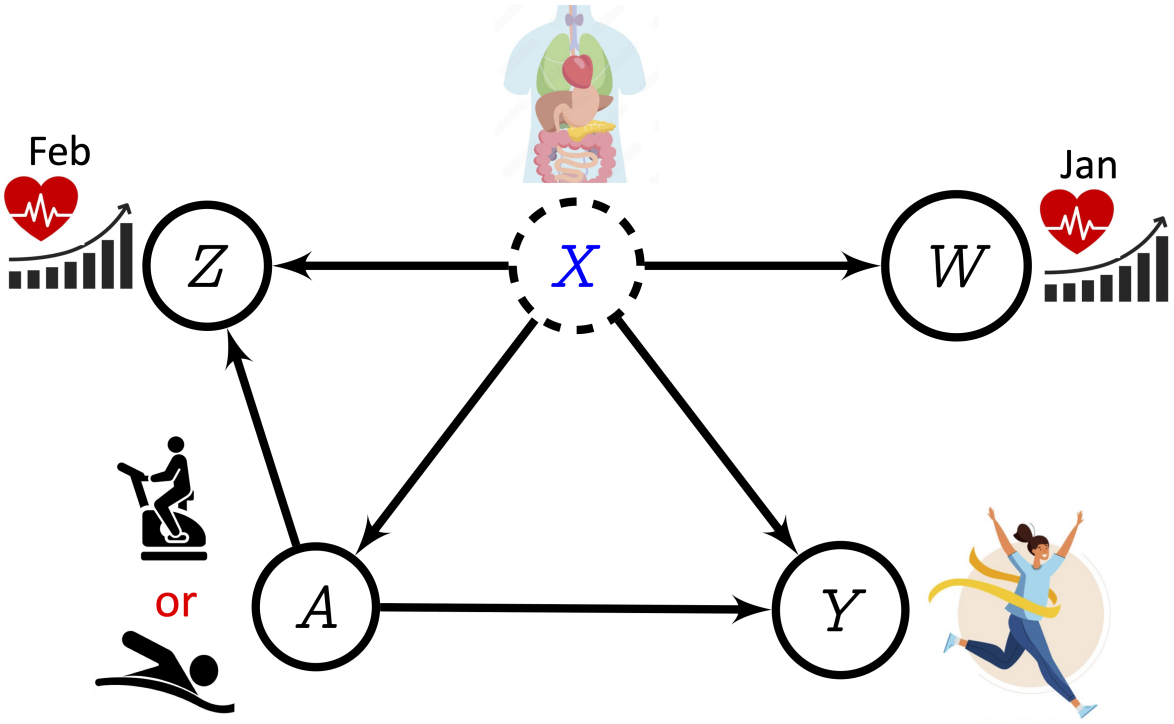


What are proxies, and when are they useful?

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



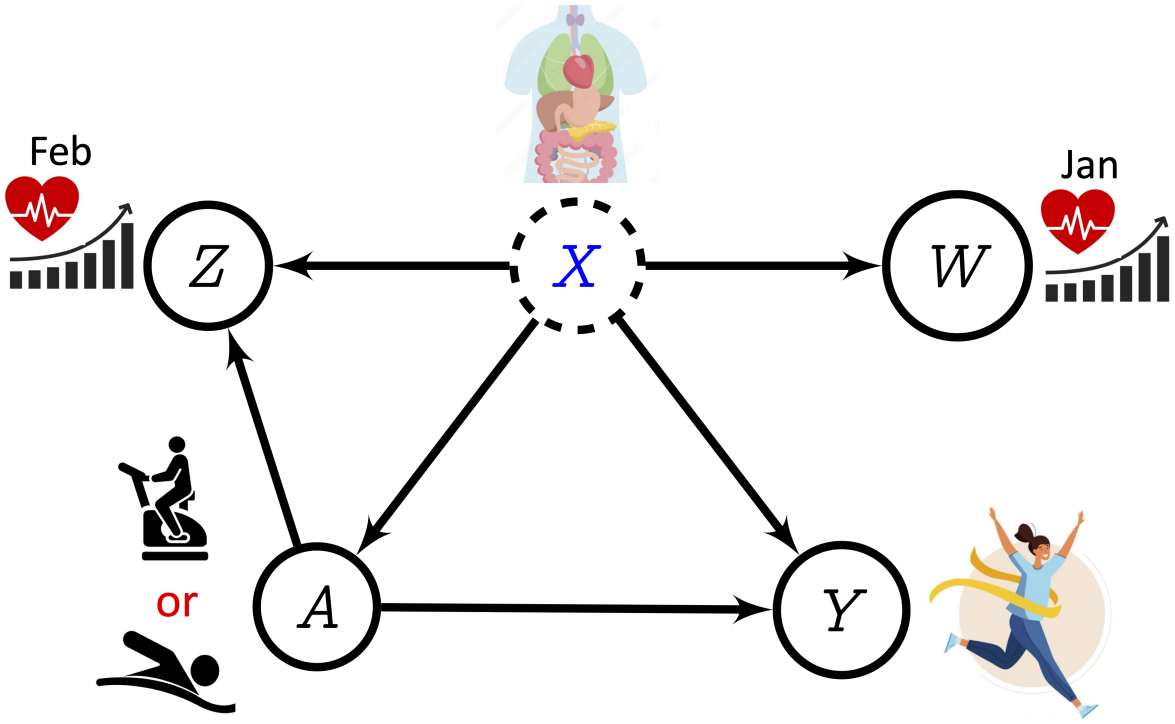
$\Longrightarrow$ Can recover $\mathbb{E}(Y^{(a)})$ from observational data

What are proxies, and when are they useful?

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

In this example:

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



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

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

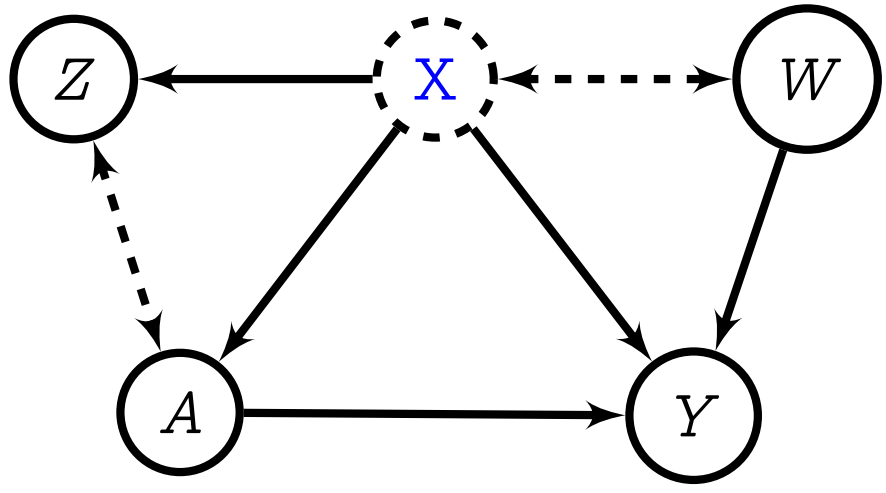
$$\mathbb{E}(Y^{(\pi(A|X))})$$

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



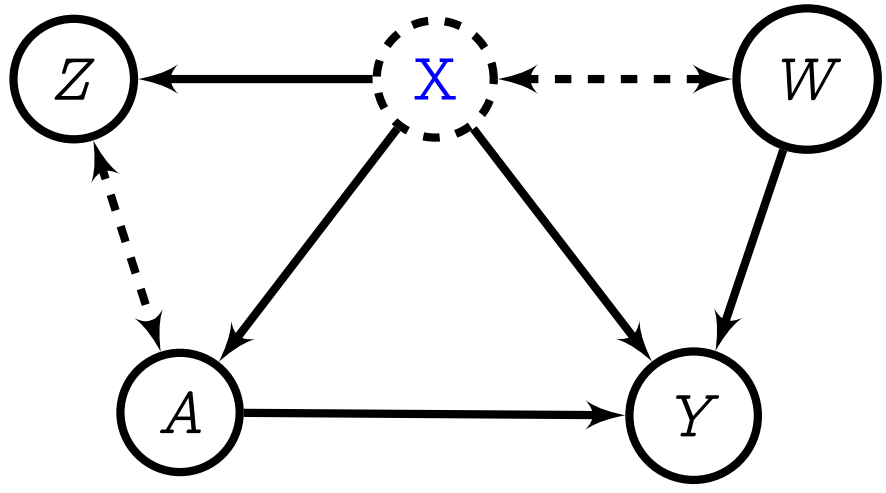
Miao, Geng, Tchetgen Tchetgen (2018): Identifying causal effects with proxy variables of an unmeasured confounder.

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:

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

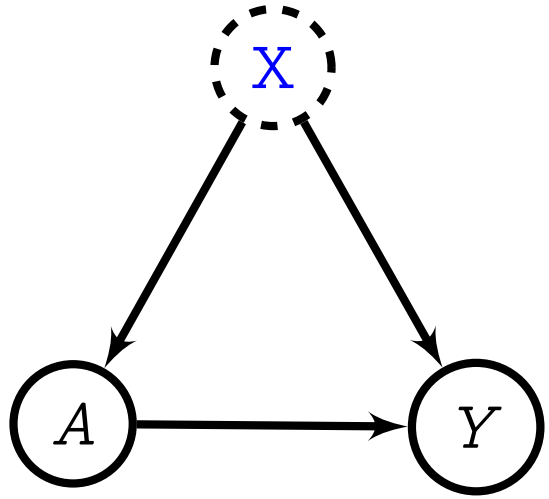
$$Y \perp\!\!\!\perp Z | (A, X)$$

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



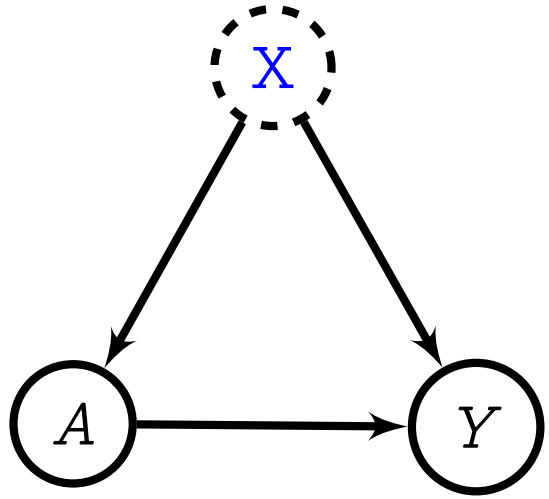
If X were observed,

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

Why proxy variables? A simple proof

The definitions are:

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



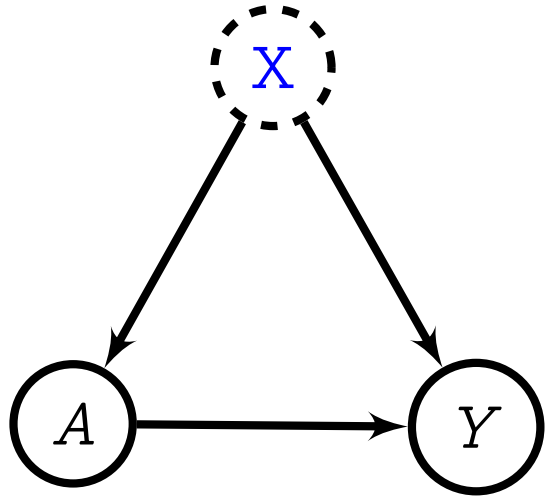
If X were observed,

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

Why proxy variables? A simple proof

The definitions are:

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



If X were observed,

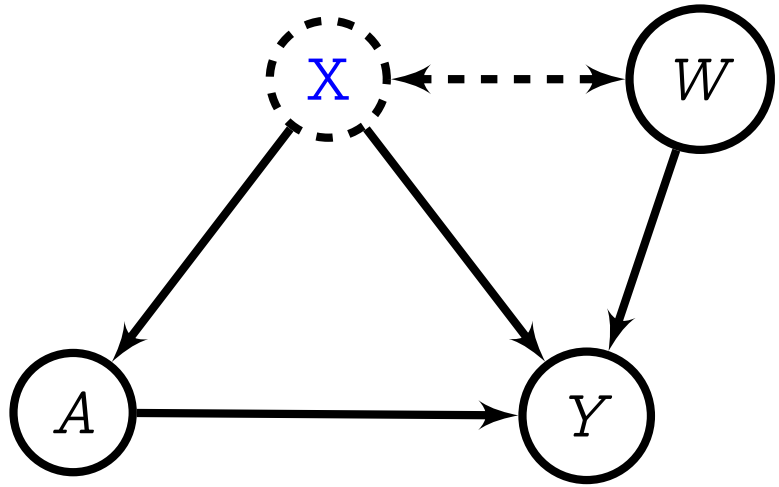
$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y|x_i, a)P(x_i) = \underbrace{P(Y|X, a)}_{d_y \times d_x}\underbrace{P(X)}_{d_x \times 1}$$

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



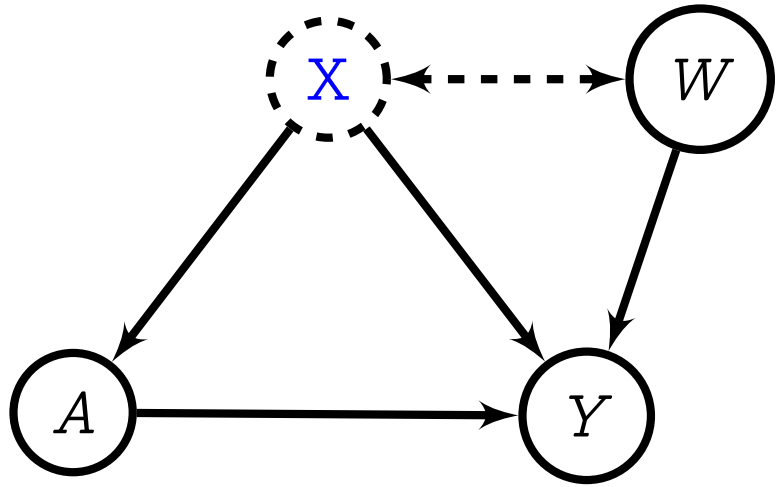
For each a, if we could solve:

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

...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:

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

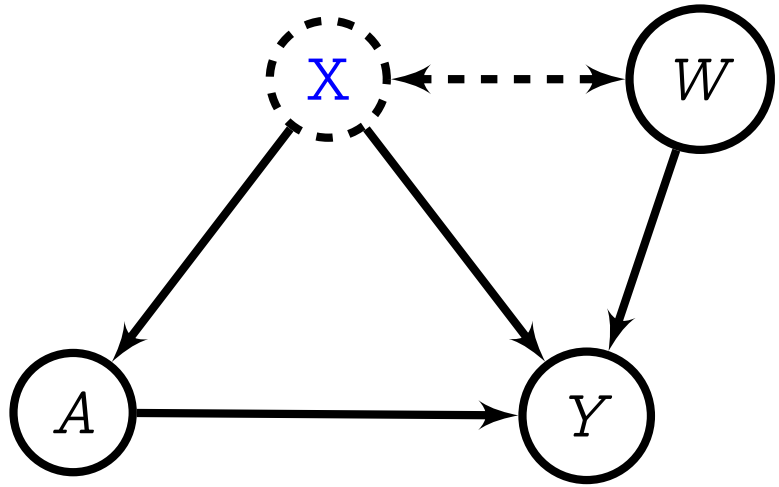
.....then

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

...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:

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

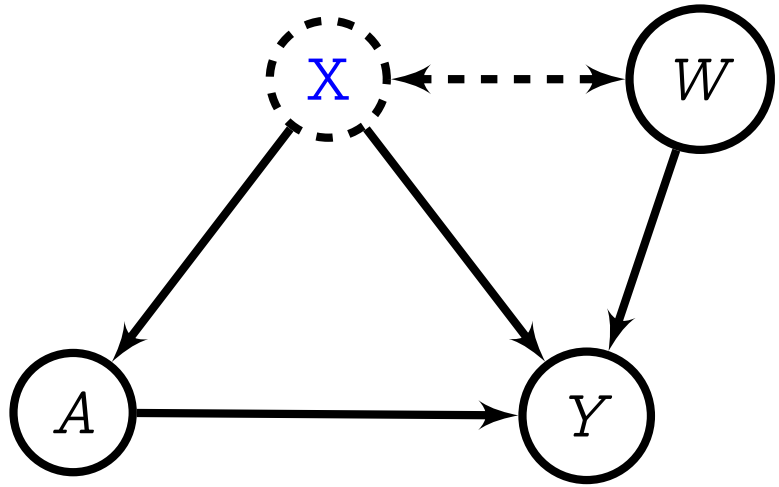
.....then

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

...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:

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

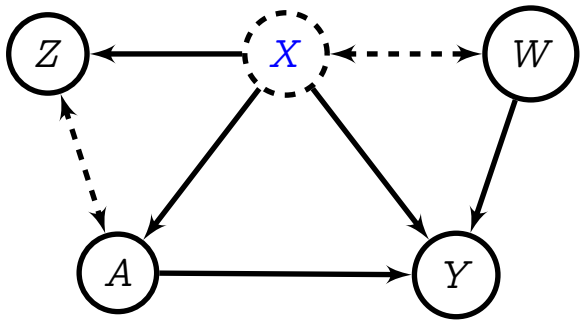
.....then

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

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

From last slide,

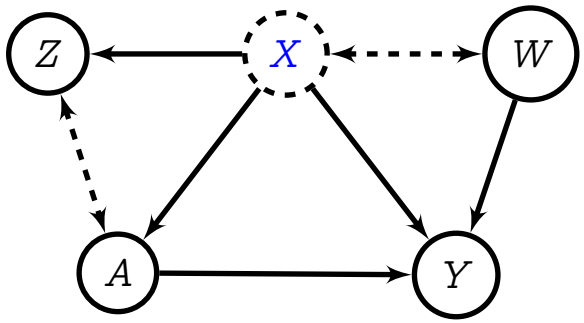
$$P(Y|X,a) = H_{w,a} P(W|X)$$



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

From last slide,

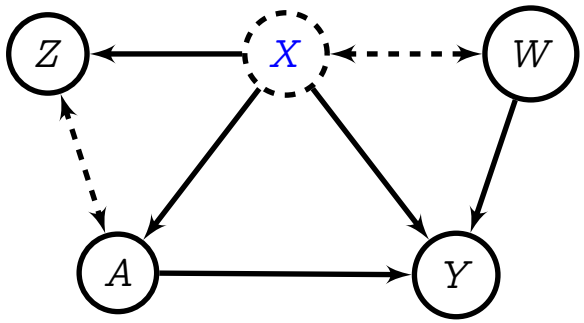
$$P(Y|X,a)\underbrace{p(X|Z,a)}_{d_x \times d_z} = H_{w,a} P(W|X) \underbrace{p(X|Z,a)}_{d_x \times d_z}$$



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

From last slide,

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



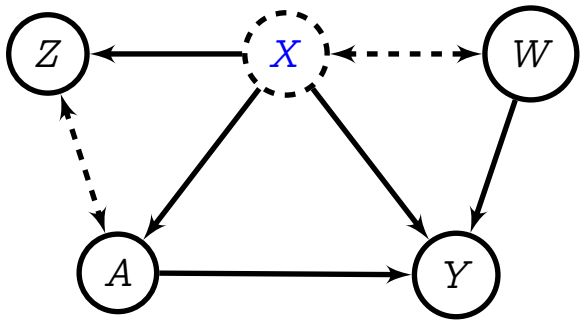
Because $W \perp\!\!\!\perp (Z,A)|X$,

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

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

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

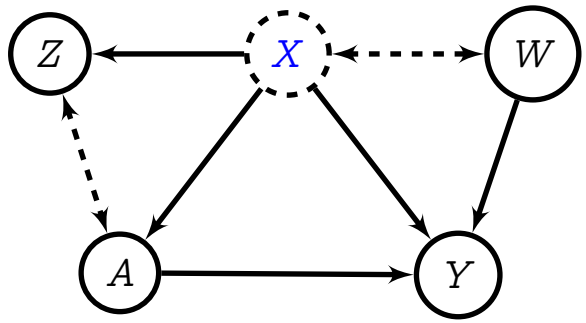
Because $Y \perp\!\!\!\perp Z|(A,X)$,

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

...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 $H_{w,a}$:

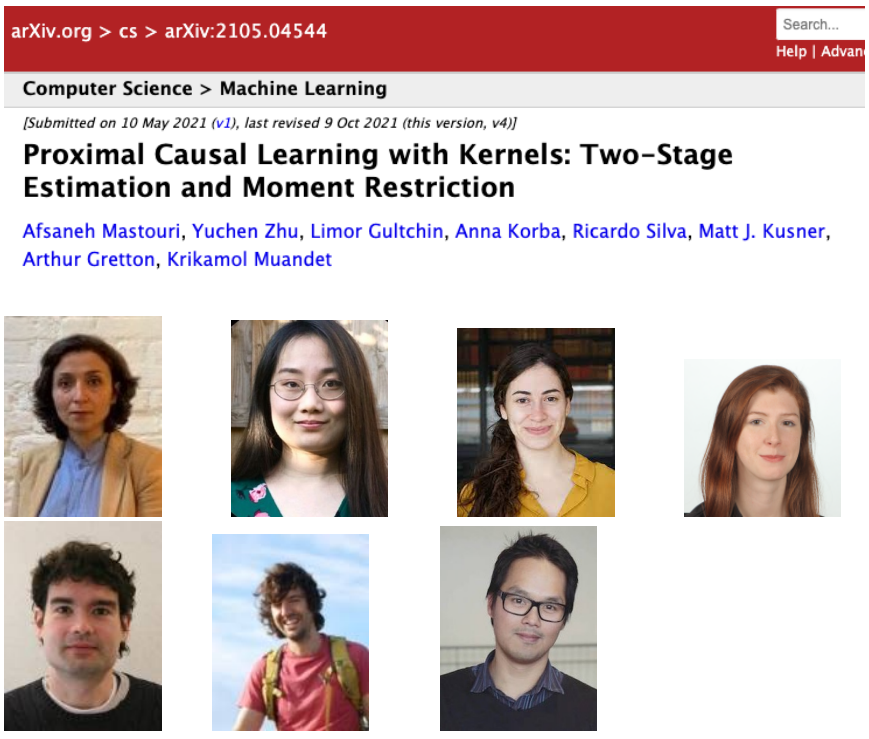
$$P(Y|Z,a) = H_{w,a} P(W|Z,a)$$

Everything observed!

Proxy/Negative Control Methods in the Real World

Unobserved confounders: proxy methods

Kernel features (ICML 2021):



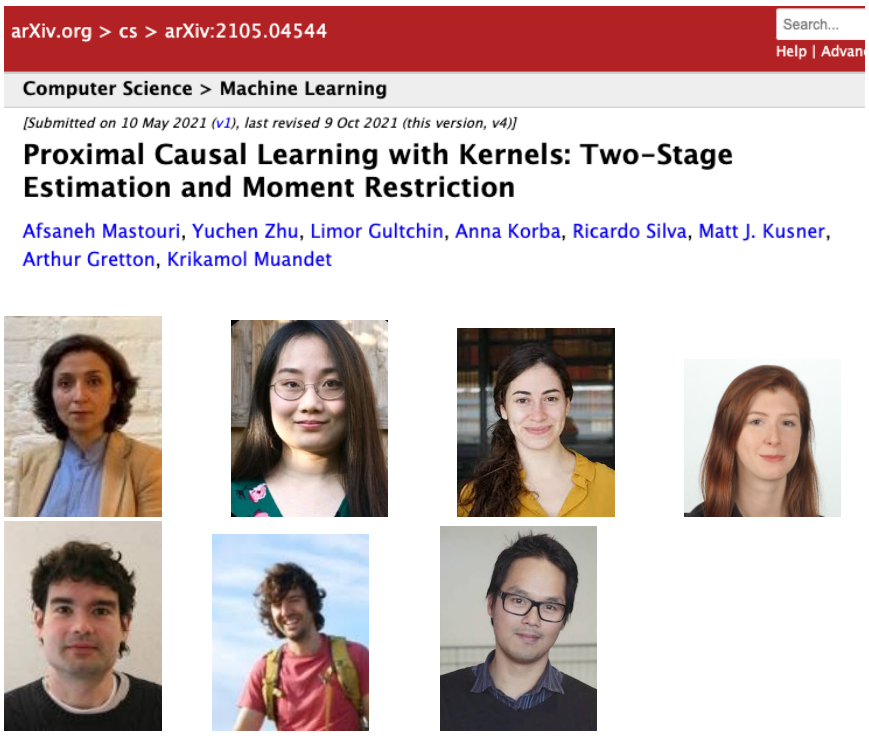
NN features (NeurIPS 2021):

Code for NN and kernel proxy methods:

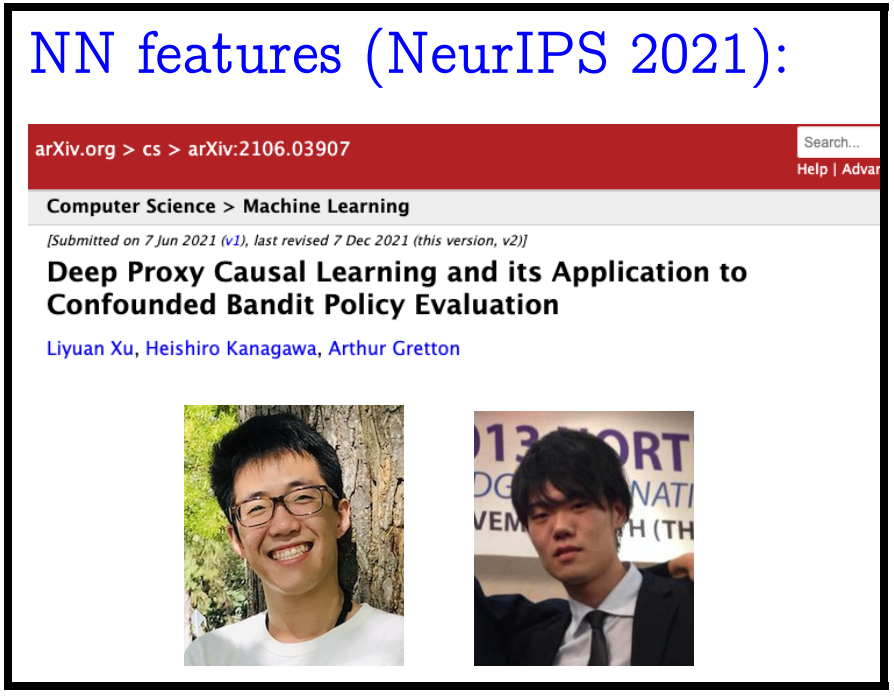
https://github.com/liyuan9988/DeepFeatureProxyVariable/

Unobserved confounders: proxy methods

Kernel features (ICML 2021):



NN features (NeurIPS 2021):



Code for NN and kernel proxy methods:

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

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 (dose-response curve)

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

....but we do not observe X.

Proxy relation, general domains

If X were observed, we would write (dose-response curve)

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

....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. Link existence requires $\diamondsuit$, identification of ATE requires $\triangle$ (and further technical assumptions) [XKG: Asspumption 2, Prop. 1,Corr. 1; Deaner]

$$\mathbb{E}[f(X)|A=a, Z=z] = 0,\ \forall(z,a) \iff f(X) = 0,\ \mathbb{P}_X \text{ a.s.} \quad \triangle$$

$$\mathbb{E}[f(X)|A=a, W=w] = 0,\ \forall(w,a) \iff f(X) = 0,\ \mathbb{P}_X \text{ a.s.} \quad \diamondsuit$$

Proxy relation, general domains

If X were observed, we would write (dose-response curve)

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

....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*.

Dose-response curve via $p(w)$:

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

Proxy relation, general domains

If X were observed, we would write (dose-response curve)

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

....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*.

Dose-response curve via $p(w)$:

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

Challenge: need a loss function for h_y

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

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

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

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

...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 $h_y(a, w)$

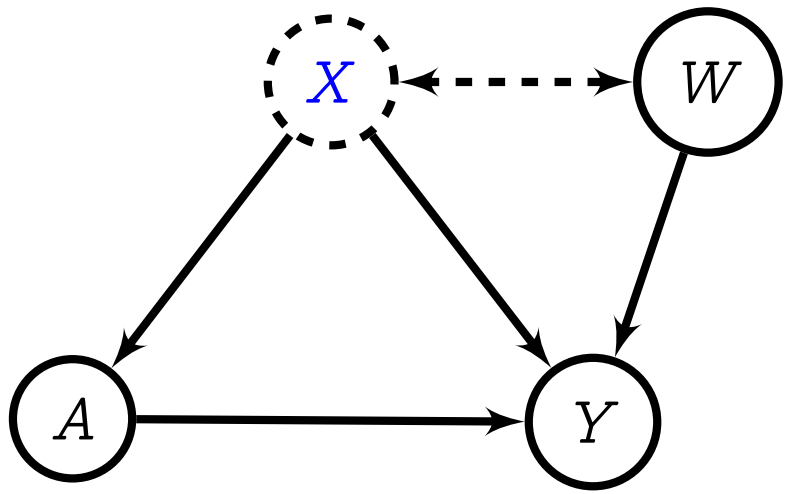
The link function is a function of two arguments

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

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 $h_y(a, w)$

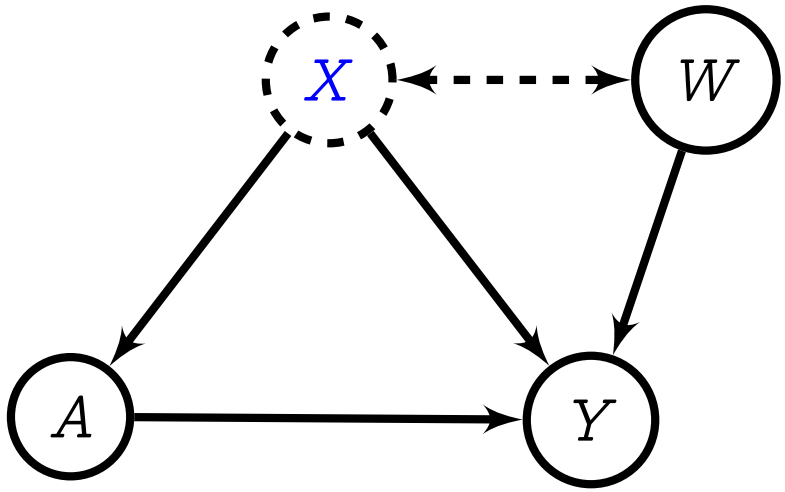
The link function is a function of two arguments

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

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:

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

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

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

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:

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

Recall link function

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

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:

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

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

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

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 $\varphi_\zeta(Z)$ 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 $\varphi_\theta, \varphi_\zeta$.

Xu, Kanagawa, G. (2021).

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

Secondary regression: learn NN features $\varphi_\zeta(Z)$ 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 $\varphi_\theta, \varphi_\zeta$.

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)

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 $\hat{F}_{\theta,\zeta}\varphi_\zeta(A, Z)$ and $\varphi_\xi(A)$

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 $\hat{F}_{\theta,\zeta}\varphi_\zeta(A, Z)$ and $\varphi_\xi(A)$
- Primary: Gradient steps on θ, ξ (backprop through Cholesky)
  - $\hat{F}_{\theta,\zeta}$ remains optimal wrt current φ_θ.

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 $\hat{F}_{\theta,\zeta}\varphi_\zeta(A, Z)$ 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

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 $\hat{F}_{\theta,\zeta}\varphi_\zeta(A, Z)$ 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:

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

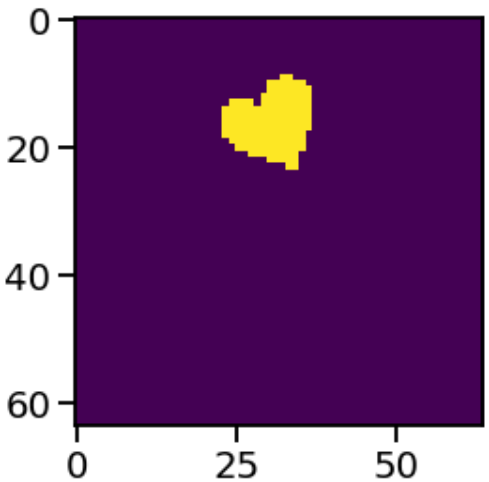
Contrast with autoencoders/sampling: must reconstruct/sample all of W.

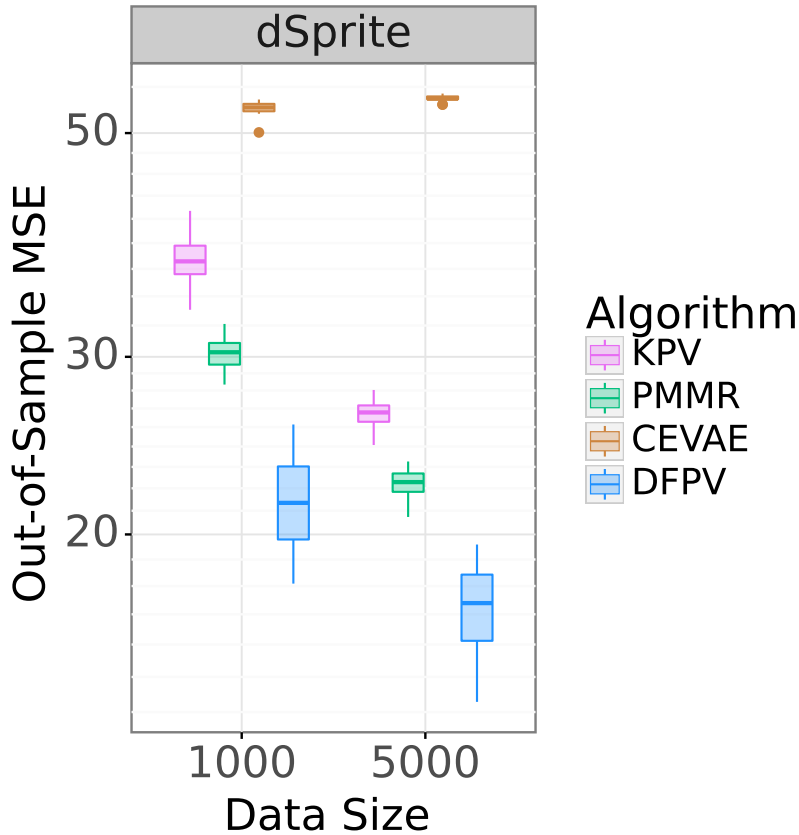
Experiments

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 $\texttt{posY}$.
- $Z = \{\texttt{scale}, \texttt{rotation}, \texttt{posX}\}$, $W =$ noisy image sharing $\texttt{posY}$
- Comparison with CEVAE (Louzios et al. 2017)



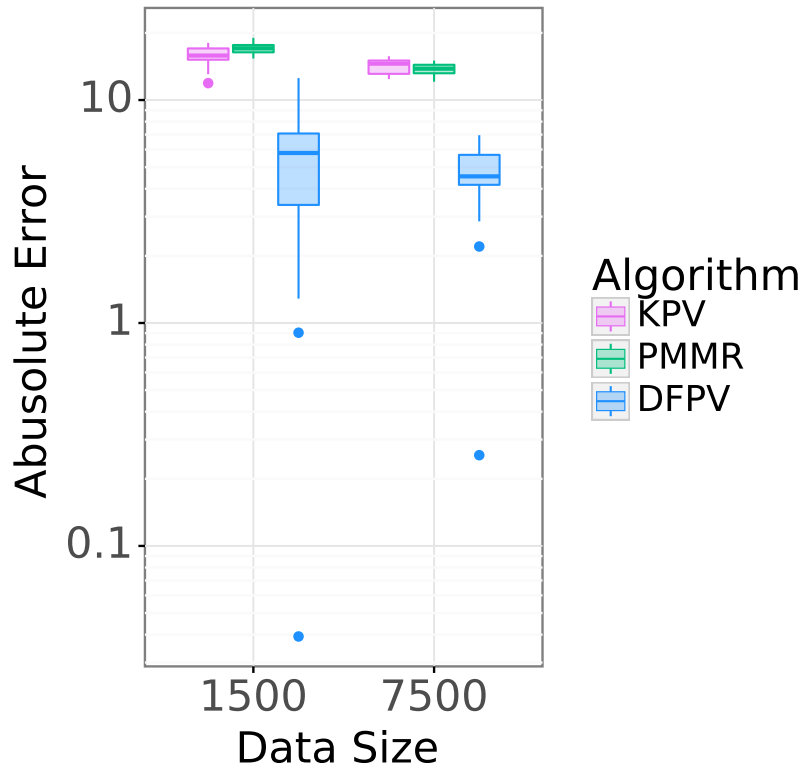


Louizos, Shalit, Mooij, Sontag, Zemel, Welling, Causal Effect Inference with Deep Latent-Variable Models (2017)

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.



Conclusions

Neural net and kernel solutions:

- ...for ATE, CATE, dynamic treatment effects
- ...even for unobserved covariates/confounders (IV and proxy methods)
- ...with treatment A, covariates X, V, proxies/instruments (W, Z) multivariate, "complicated"
- Convergence guarantees for kernels and NN

Key messages:

- Don't ~~meet your heroes~~ model/sample hidden variables
- "Ridge regression is all you need"

Code available for all methods

Research support

Work supported by:

The Gatsby Charitable Foundation

Google Deepmind

Google DeepMind

Questions?


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