Learning problems from the Causal Hierarchy

Ciarán Gilligan-Lee

Spotify & University College London

Motivating example I

Motivating example

- The findings were very weird indeed, flying in the face of medical knowledge and confounding experts

- Yet the finding was irrefutable: if you smoked the data said you were less at risk of COVID-19

Should we all start smoking?

See *Collider bias undermines our understanding of COVID-19 disease risk and severity*, Griffith et al.

Motivating example

- At the start of the pandemic, only healthcare workers (who smoke less) and people with severe COVID-19 symptoms were tested.

- Smokers with no COVID-19 symptoms were massively under represented in the observed data.

- Hence, of those tested, the non-smokers are more likely to have COVID-19 than smokers.

Motivating example

- In our example any action based on these correlations such as how patients with or with COVID-19 who smoke are treated would not increase patient survival.

Take home: Relying on correlations extracted from observational data can lead to embarrassing, costly, and dangerous mistakes.

- To overcome this, we need to understand cause and effect

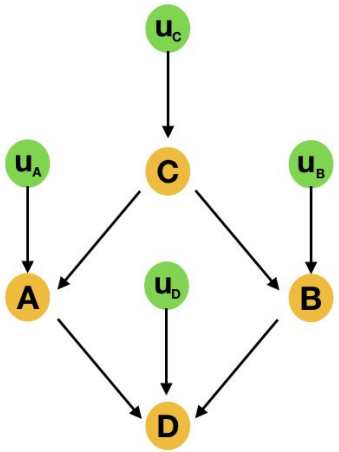
How do we understand cause and effect?

- Standard approach is to run randomised control trial (A/B test)
- This would certainly help us understand the average causal effect of smoking on COVID-19 risk
- But much of the time A/B tests can’t be performed. At Spotify A/B tests could be too damaging to user experience:
  - “Do app crashes cause churn?”
  - “Does podcast consumption cause retention?”
- This is where causal inference comes in!
- But things get even more tricky when more than one action or treatment occurs at the same time...

Disentangling joint-interventions

- In many applications, only a single intervention is possible at a given time, or interventions are applied one after another in a sequential manner
- However, in some areas, **multiple interventions are concurrently applied**:
  - in medicine, patients that possess many commodities may have to be simultaneously treated with multiple prescriptions;
  - in computational advertising, people may be targeted by multiple concurrent campaigns, and so on.
  - during the pandemic, many interventions were applied at same time, e.g. mask wearing, work from home, schools closed, etc.
- What can we do in this case? First, let's define causal models and the Causal Hierarchy...

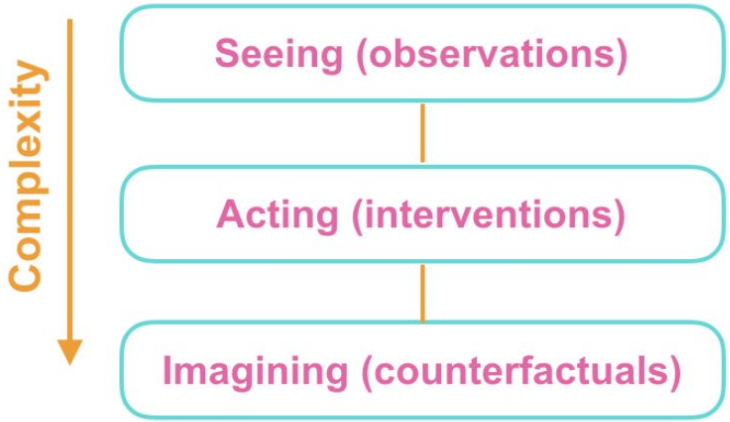
Causal Models

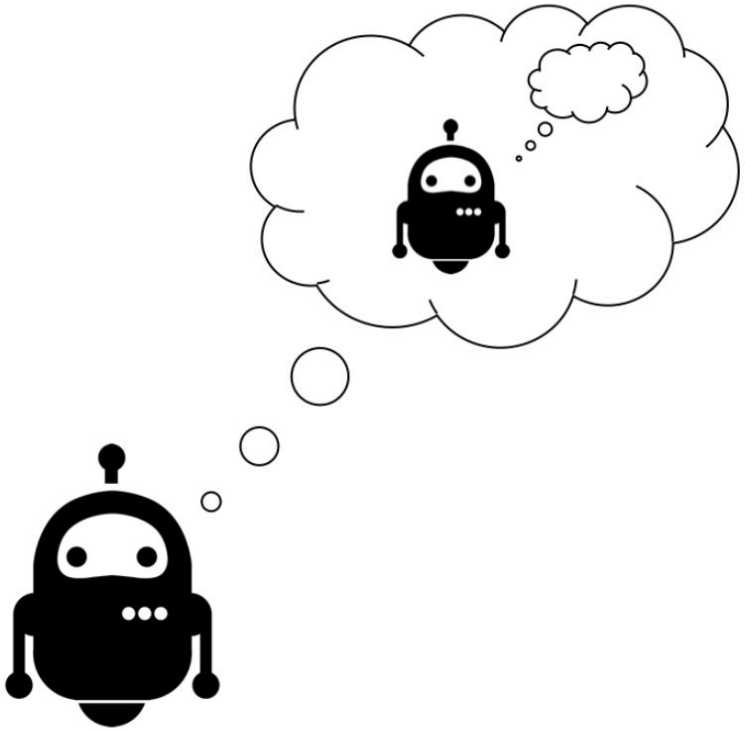


- Observed terms are deterministic function of parents and latent “noise”
- Noise terms are distributed according to latent distribution
- $A = f(C, u_A)$, $u_A \sim p(u_A)$
- These jointly generate $P(A|C)$

What can we do with them

Inference Hierarchy

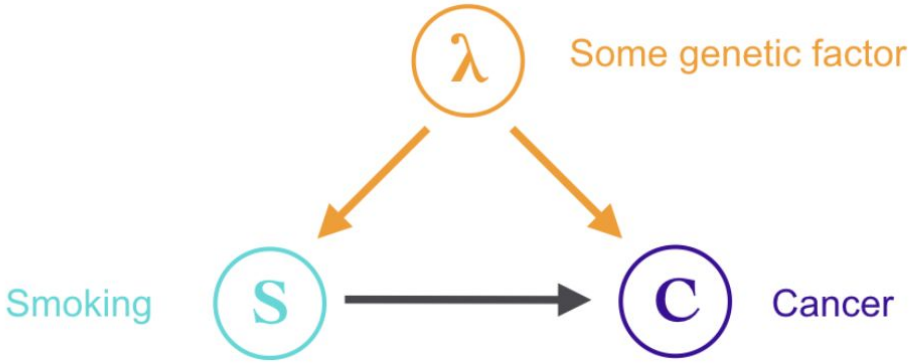




Seeing (observations)

Acting (interventions)

Imagining (counterfactuals)



given

Probability of cancer

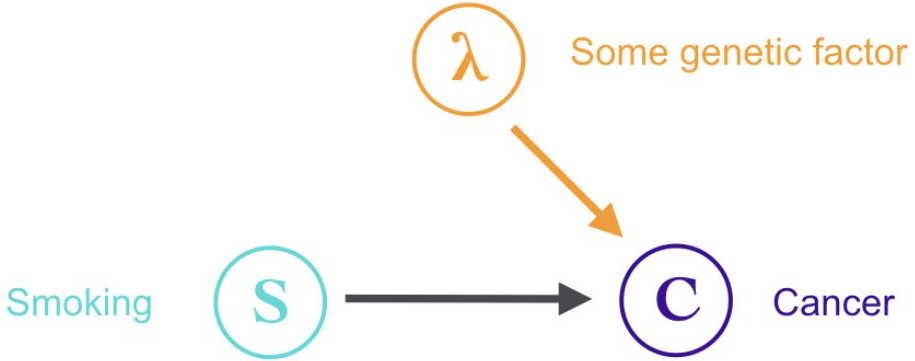
Smoking has been observed

$$P(C = T|S = T)$$

Seeing (observations)

Acting (interventions)

Imagining (counterfactuals)



given

Probability of cancer

subject was made to smoke

$$P(C = T | \mathrm{do}(S = T))$$

Simulate randomised controlled trial

Learning Problem I: disentangling interventions at 2nd layer

Given samples from observational and joint-interventions data

$$\mathbb{E}[Y|X_i = x_i, X_j = x_j, C = c], \text{ and } \mathbb{E}[Y|\text{do}(X_i = x_i, X_j = x_j), C = c]$$

When can we learn, or **identify**, conditional average causal effects of single-interventions

$$\mathbb{E}[Y|\text{do}(X_i = x_i), X_j = x_j, C = c], \text{ or } \mathbb{E}[Y|X_i = x_i, \text{do}(X_j = x_j), C = c]$$

Learning Problem I: disentangling interventions at 2nd layer

Given samples from observational and joint-interventions data

$$\mathbb{E}[Y|X_i = x_i, X_j = x_j, C = c], \text{ and } \mathbb{E}[Y|\text{do}(X_i = x_i, X_j = x_j), C = c]$$

When can we learn, or **identify**, conditional average causal effects of single-interventions

$$\mathbb{E}[Y|\text{do}(X_i = x_i), X_j = x_j, C = c], \text{ or } \mathbb{E}[Y|X_i = x_i, \text{do}(X_j = x_j), C = c]$$

Identifiability

A quantity is identifiable from a specific type of data if every model that agrees on that data produces the same value for the quantity

Hence, if two models agree on the data, but not on the quantity, then it is not identifiable from that data

Our quantity is non-Identifiable in general...

$\mathcal{M}$	$\mathcal{M}'$
$X_1 = U_1$	$X_1 = U_1$
$X_2 = X_1 U_2$	$X_2 = U_2$
$Y = X_1 X_2 U_Y$	$Y = X_1 X_2 U_Y$

U_1 = U_2 = U_y
perfectly correlated bits

Intuition:

- All variables are binary, and all latents are perfectly correlated. So in model M, one has X_2 = X_1.U_1 = U_1.U_2 = U_2.U_2 = U_2
- So observationally, the models look the exact same! Moreover as Y is the same function of X's in both models, joint-interventions are the same
- But when we intervene on X_1, X_2 behaves differently in both models, as X_2 doesn't causally depend on X_1 in M', but it does in M.
- So observations and joint-interventions are not enough to fully constrain single-interventions. That is, we need more assumptions for identifiability

Disentangling joint-interventions: Identifiability?

$\mathcal{M}$	$\mathcal{M}'$
$X_1 = U_1$	$X_1 = U_1$
$X_2 = X_1 U_2$	$X_2 = U_2$
$Y = X_1 X_2 U_Y$	$Y = X_1 X_2 U_Y$

U_1 = U_2 = U_y
perfectly correlated bits

(a) Observational joint distribution.

$\mathsf{P}(X_1, X_2, Y)$	$Y=0$	$Y=1$
$X_1, X_2 = 0, 0$	$1-p$	0
$X_1, X_2 = 0, 1$	0	0
$X_1, X_2 = 1, 0$	0	0
$X_1, X_2 = 1, 1$	0	p

(b) Joint interventional distribution.

$\mathsf{P}(Y \mid \mathrm{do}(X_1, X_2))$	$Y=0$	$Y=1$
$\mathrm{do}(X_1 = 0, X_2 = 0)$	1	0
$\mathrm{do}(X_1 = 0, X_2 = 1)$	1	0
$\mathrm{do}(X_1 = 1, X_2 = 0)$	1	0
$\mathrm{do}(X_1 = 1, X_2 = 1)$	$1-p$	p

(c) Interventional distribution on X_2.

$\mathsf{P}(Y, X_1 \mid \mathrm{do}(X_2))$		$Y=0$	$Y=1$
$\mathrm{do}(X_2 = 0)$	$X_1 = 0$	$1-p$	0
	$X_1 = 1$	p	0
$\mathrm{do}(X_2 = 1)$	$X_1 = 0$	$1-p$	0
	$X_1 = 1$	0	p

It *is* idenitifiable from *extra assumptions*

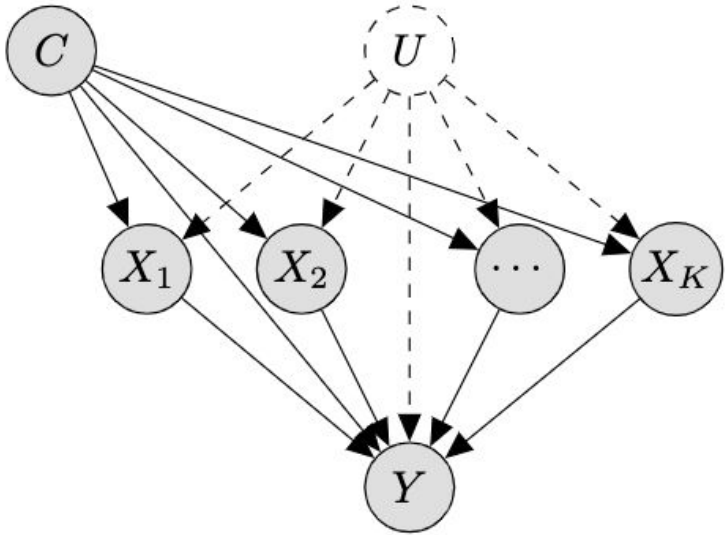
Theorem 2 (Identifiability of disentangled conditional average treatment effects in additive noise models with symmetric structure)**.**

Let $\mathcal{M} = \langle \{\boldsymbol{C}, \boldsymbol{X}, Y\}, \boldsymbol{U}, \boldsymbol{f}, \mathsf{P}_U \rangle$ *be an SCM, where*

$$X_i = f_i(\boldsymbol{C}) + U_i, \quad \forall i = 1, \ldots, K,$$
$$Y = f_Y(\boldsymbol{C}, \boldsymbol{X}) + U_Y,$$

$C \perp\!\!\!\perp U$, *and* $\mathsf{P}_U \sim \mathcal{N}(0, \Sigma)$. *The estimand* $\mathbb{E}[Y|do(X_i), C]$ *is identifiable from the conjunction of two data regimes:*

1. *the observational distribution,*
2. *any interventional distribution on a set of treatments* $\boldsymbol{X}_{int} \subseteq \boldsymbol{X}$ *that holds* X_i*:* $X_i \in \boldsymbol{X}_{int}$.



It is idenitifiable from extra assumptions

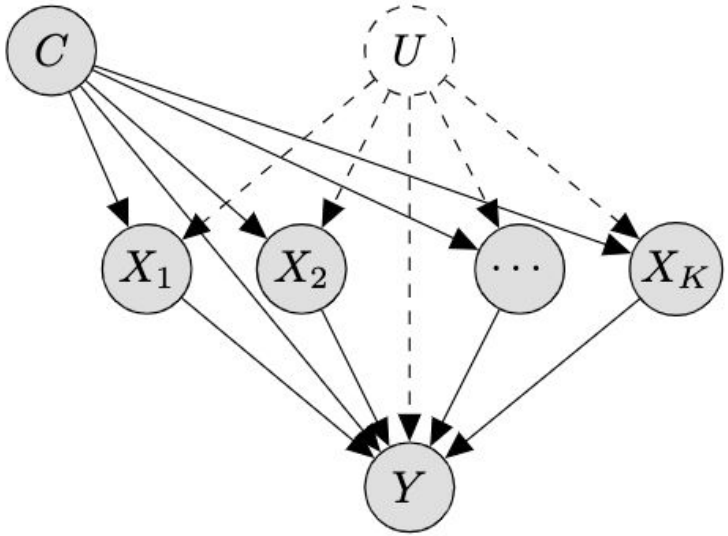
Theorem 2 (Identifiability of disentangled conditional average treatment effects in additive noise models with symmetric structure)**.**

Let $\mathcal{M} = \langle \{\boldsymbol{C}, \boldsymbol{X}, Y\}, \boldsymbol{U}, \boldsymbol{f}, \mathsf{P}_U \rangle$ *be an SCM, where*

$$X_i = f_i(\boldsymbol{C}) + U_i, \quad \forall i = 1, \ldots, K,$$
$$Y = f_Y(\boldsymbol{C}, \boldsymbol{X}) + U_Y,$$

$C \perp\!\!\!\perp U$, *and* $\mathsf{P}_U \sim \mathcal{N}(0, \Sigma)$. *The esti[illegible]d* $\mathbb{E}[Y|do(X_i), C]$ *is identifiable from the conjunction of tw[illegible] data regimes:*

1. *the observational distribution,*
2. *any interventional distribution on a set of treatments* $\boldsymbol{X}_{int} \subseteq \boldsymbol{X}$ *that holds* X_i*:* $X_i \in \boldsymbol{X}_{int}$.



This additive noise model still allows for correlations and interactions between treatments, through observed and unobserved confounders

Can these assumptions be weakened?

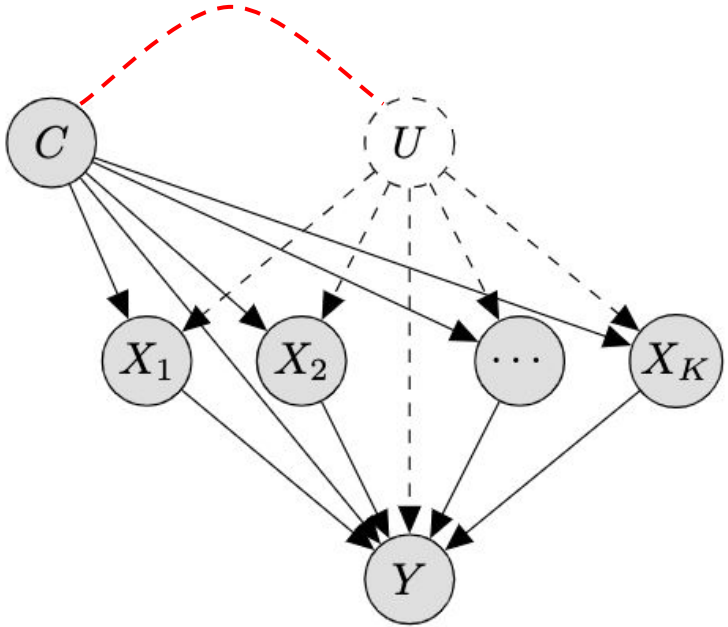
Theorem 2 (Identifiability of disentangled conditional average treatment effects in additive noise models with symmetric structure)**.**

Let $\mathcal{M} = \langle \{\boldsymbol{C}, \boldsymbol{X}, Y\}, \boldsymbol{U}, \boldsymbol{f}, \mathsf{P}_U \rangle$ *be an SCM, where*

$$X_i = f_i(\boldsymbol{C}) + U_i, \quad \forall i = 1, \ldots, K,$$
$$Y = f_Y(\boldsymbol{C}, \boldsymbol{X}) + U_Y,$$

and $\mathsf{P}_U \sim \mathcal{N}(0, \Sigma)$*. The estimand* $\mathbb{E}[Y|do(X_i), C]$ *is identifiable from the conjunction of two data regimes:*

1. *the observational distribution,*
2. *any interventional distribution on a set of treatments* $\boldsymbol{X}_{int} \subseteq \boldsymbol{X}$ *that holds* X_i*:* $X_i \in \boldsymbol{X}_{int}$*.*



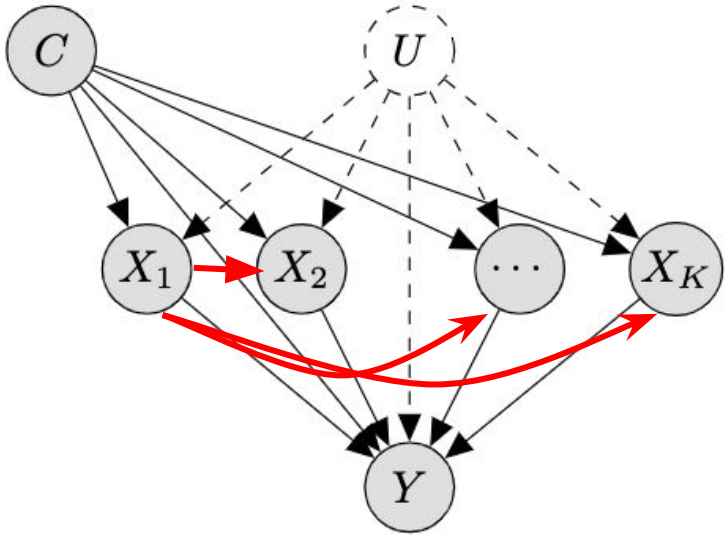
Can these assumptions be weakened?

Theorem 1 (Identifiability of disentangled conditional average treatment effects in additive noise models with a causal dependency between treatments)**.** *Let* $\mathcal{M} = \langle \{\boldsymbol{C}, \boldsymbol{X}, Y\}, \boldsymbol{U}, \boldsymbol{f}, \mathsf{P}_U \rangle$ *be an SCM, where*

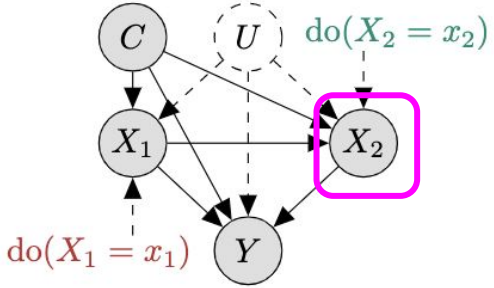
$$
\begin{aligned}
X_i &= f_i(\boldsymbol{C}) + U_i \\
X_j &= f_j(\boldsymbol{C}, X_i) + U_j \\
Y &= f_Y(\boldsymbol{C}, \boldsymbol{X}) + U_Y,
\end{aligned}
$$

and $\mathsf{P}_U \sim \mathcal{N}(0, \Sigma)$*. The estimand* $\mathbb{E}[Y|do(X_j), C]$ *is identifiable from the conjunction of two data regimes:*

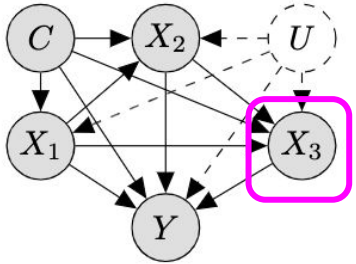
1. *the observational distribution,*
2. *the joint interventional distribution on* (X_i, X_j)*.*



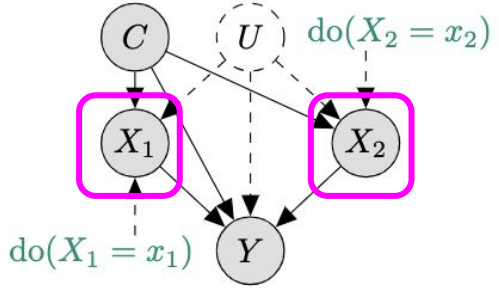
In summary...



(a) Only $\mathbb{E}[Y|C = c, \text{do}(X_2 = x_2)]$ is generally identifiable.



(b) Only $\mathbb{E}[Y|C = c, \text{do}(X_3 = x_3)]$ is generally identifiable.



(c) All $\mathbb{E}[Y|C = c, \text{do}(X_i = x_i)]$ are generally identifiable.

Figure 2: Causal Graphs illustrating under which conditions the single-variable causal effect on the outcome $\mathbb{E}[Y|C = c, \text{do}(X_i = x_i)]$ is identifiable from the observational and joint interventional data regimes.

Learning algorithm based on results

Estimating an SCM from a combination of observational and interventional data boils down to:

1. estimating the structural equations, and
2. estimating the noise distribution

$$\mathbb{E}[Y|\boldsymbol{C}; \mathrm{do}(\boldsymbol{X}_{\mathrm{int}}); \boldsymbol{X}_{\mathrm{obs}}] = f_Y(\boldsymbol{C}; \boldsymbol{X}) + \mathbb{E}[U_Y|\boldsymbol{X}_{\mathrm{obs}}].$$

We employ an Expectation-Maximisation-style iterative algorithm to achieve this

$$L(x_i; \theta, \Sigma) = \mathsf{P}_U(x_i - f_i(\mathrm{PA}(x_i); \theta); \Sigma)$$

Algorithm 1 SCM Estimation for Symmetric ANMs

Input: Dataset $\mathcal{D}$
Output: Parameter estimates $\widehat{\theta}$, $\widehat{\Sigma}$
1: Initialise $\widehat{\theta}$ and $\widehat{\Sigma}$
2: **while** not converged **do**
3: // Solve for θ with fixed $\widehat{\Sigma}$
4: Optimise log-likelihood in Eq. 7
5: // Solve for Σ with fixe[illegible] $\widehat{\theta}$
6: Estimate $\widehat{\Sigma}$ from $\widehat{U} = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}; \theta)$
7: **return** $\widehat{\theta}, \widehat{\Sigma}$

Full details in the paper…

Presented at NeurIPS Causal Inference & Machine Learning workshop and will be on arXiv soon!

Disentangling causal effects from sets of interventions in the presence of unobserved confounders

Olivier Jeunen[2] **Ciarán M. Gilligan-Lee** **Rishabh Mehrotra** **Mounia Lalmas**
Spotify, London, UK

Abstract

The ability to answer causal questions is crucial in many domains, as causal inference allows one to understand the impact of interventions. In many applications, only a single intervention is possible at a given time. However, in some important areas, multiple interventions are concurrently applied. Disentangling the effects of single interventions from jointly applied interventions is a challenging task—especially as simultaneously applied interventions can interact. This problem is made harder still by unobserved confounders, which influence both treatments and outcome. We address this challenge by aiming to learn the effect of a single-intervention from both observational data and sets of interventions. We prove that this is not generally possible, but provide identification proofs demonstrating that it can be achieved in certain classes of additive noise models—even in the presence of unobserved confounders. Importantly, we show how to incorporate observed covariates and learn heterogeneous treatment effects conditioned on them for single-interventions.

1 INTRODUCTION

manner. However, in some important areas, multiple interventions are concurrently applied. For instance, in medicine, patients that possess many commodities may have to be simultaneously treated with multiple prescriptions; in computational advertising, people may be targeted by multiple concurrent campaigns; and in dietetics, the nutritional content of meals can be considered a joint intervention from which we wish to learn the effects of individual nutritional components.

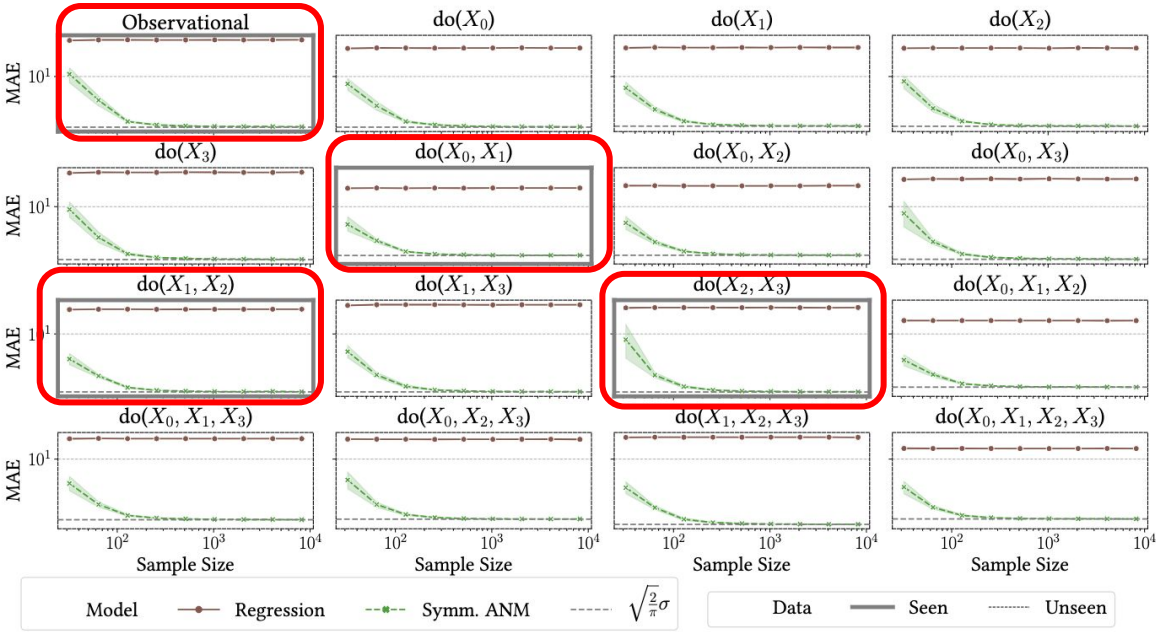
Disentangling the effects of single interventions from jointly applied interventions is a challenging task—especially as simultaneously applied interventions can interact, leading to consequences not seen when considering single interventions separately. This problem is made harder still by the possible presence of unobserved confounders, which influence both treatments and outcome. This paper addresses this challenge, by aiming to learn the effect of a single-intervention from both observational data and sets of interventions. We prove that this is not generally possible, but provide identification proofs demonstrating it can be achieved in certain classes of non-linear causal models with additive Gaussian noise—even in the presence of unobserved confounders. Importantly, we show how to incorporate observed covariates, which can be high-dimensional, by learning heterogeneous treatment effects conditioned on them for single-interventions.

Our main contributions are:

1. A proof that without restrictions on the causal

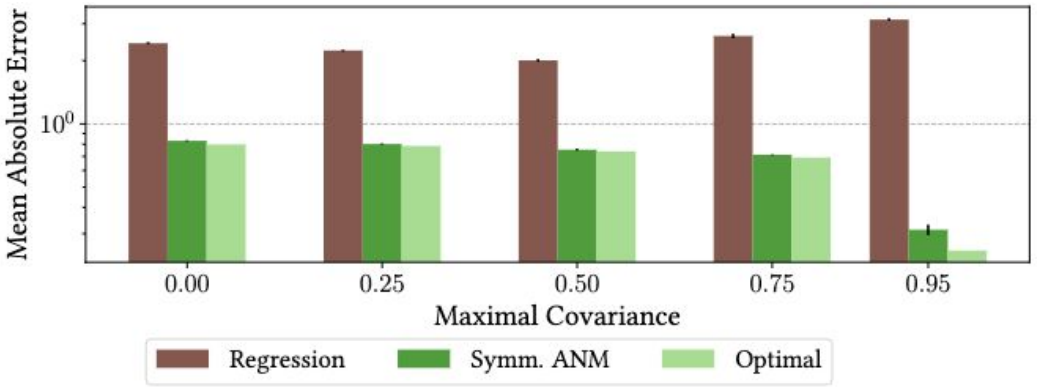
Experiments: How does our method compare to just performing regression?

As amount of data increases, our method can accurately disentangle joint interventions even in the presence of unobserved confounding



Experiments: How robust is learning to increasing confounding strength?

Using semi-synthetic data derived from a stroke trial with multiple concurrent treatment dosages, we see our method is robust to increasing levels of unobserved confounding



Examples of disentangling problem at Spotify

- There are a range of playlists/albums/podcasts that are recommended to a user at a given time, what's the individual impact of each one?

- There are a collection of actions an artist can take to build their fanbase and improve their career, which ones have the biggest effect for a given artist?

And many, many more….

Motivating example II

- Diarrheal diseases are a leading cause of disease and mortality in the developing world
- To reduce diarrheal diseases in children in the Busia district of Western Kenya, a local NGO built protective cement structures around a randomly selected group of springs
- Researchers from Abdul Latif Jameel Poverty Action Lab (JPAL), worked with the NGO to estimate average treatment effect of intervention

Motivating example II

- They found that spring protection significantly reduces diarrhea for children under age three by 25%

- Should we scale this intervention up? Interventions are expensive, need to be certain it will help as expected

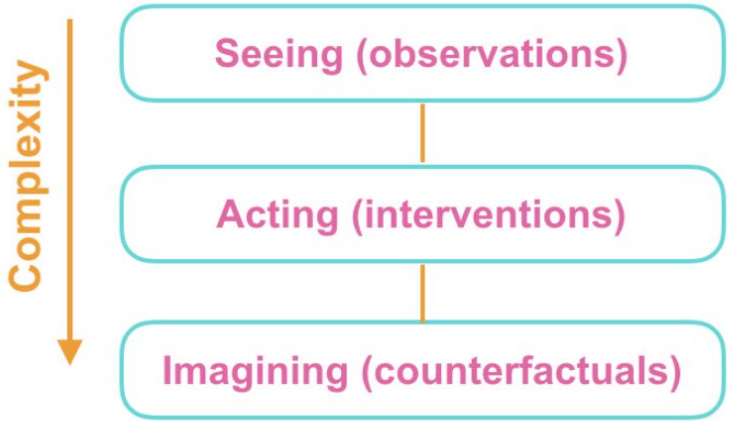
- To really answer this, need to answer “How likely is it that the negative outcome was caused by the exposure to diarrhea, and not something else?”

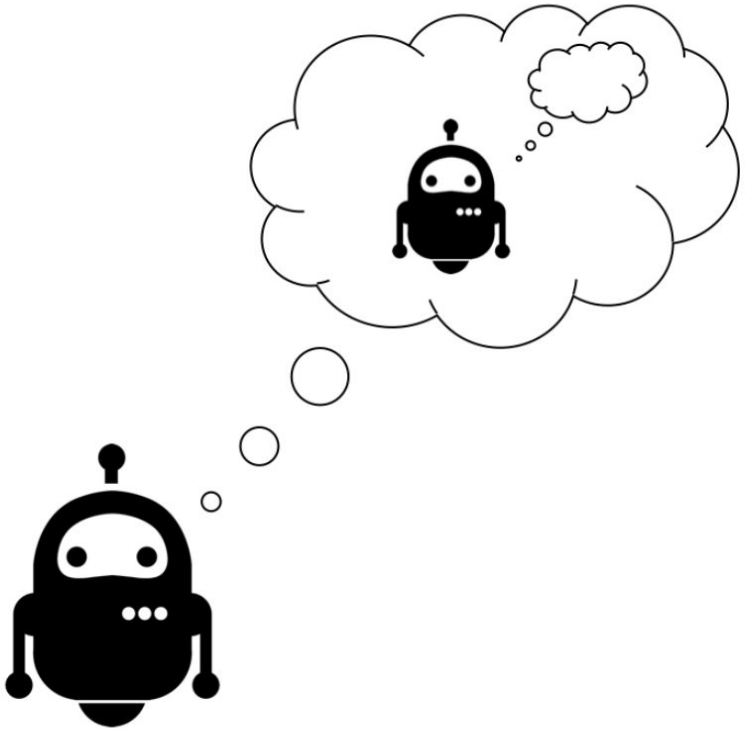
Question: “Given a child developed diarrhea after drinking from an unprotected spring, would they have still developed it if they drank from a protected spring?”

- But how do we answer this question??

What can we do with them

Inference Hierarchy

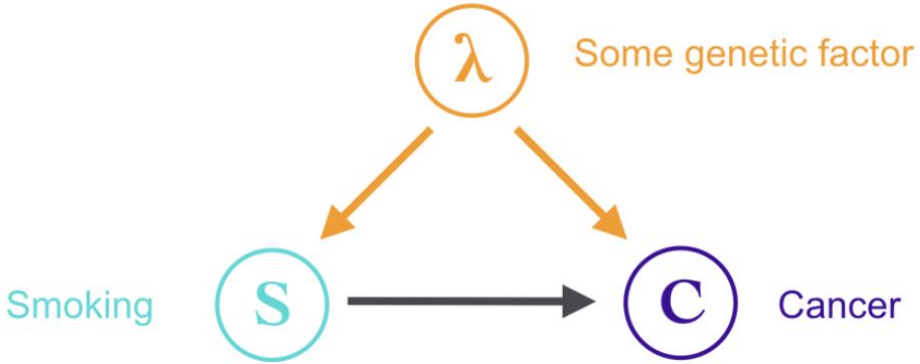




Seeing (observations)

Acting (interventions)

Imagining (counterfactuals)



Counterfactuals are "what-if" questions, very powerful in personalised decision making

given

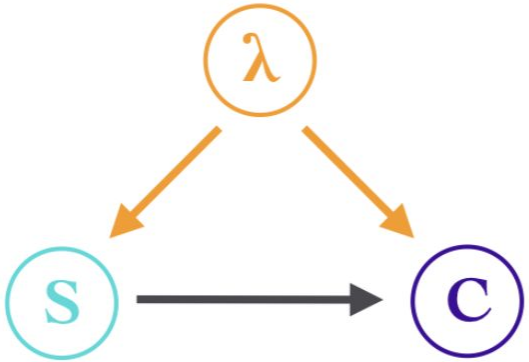
Probability of subject not having cancer

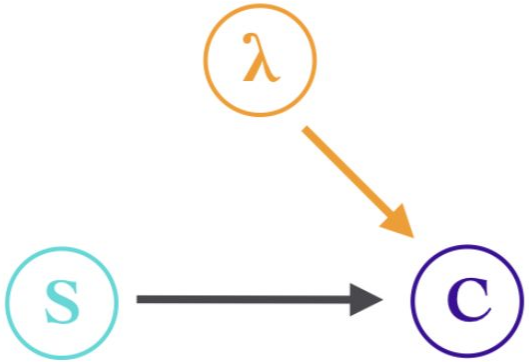
subject has cancer

and if subject was made to not smoke

$$P(C = F | C = T, \mathrm{do}(S = F))$$

"Given subject has cancer, what is the chance they wouldn't if they didn't smoke?"





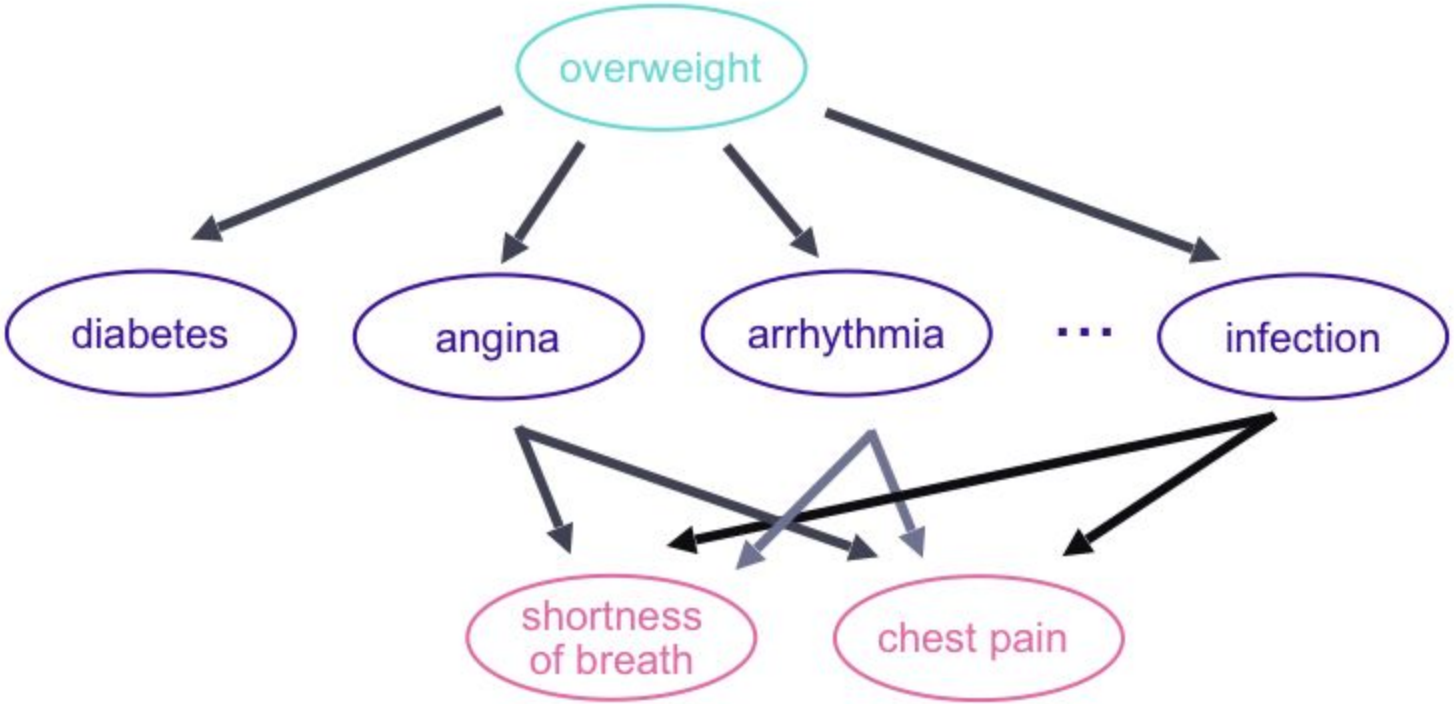
Counterfactual Inference compute P(C=F | C=T, S=T, do(S=F)):

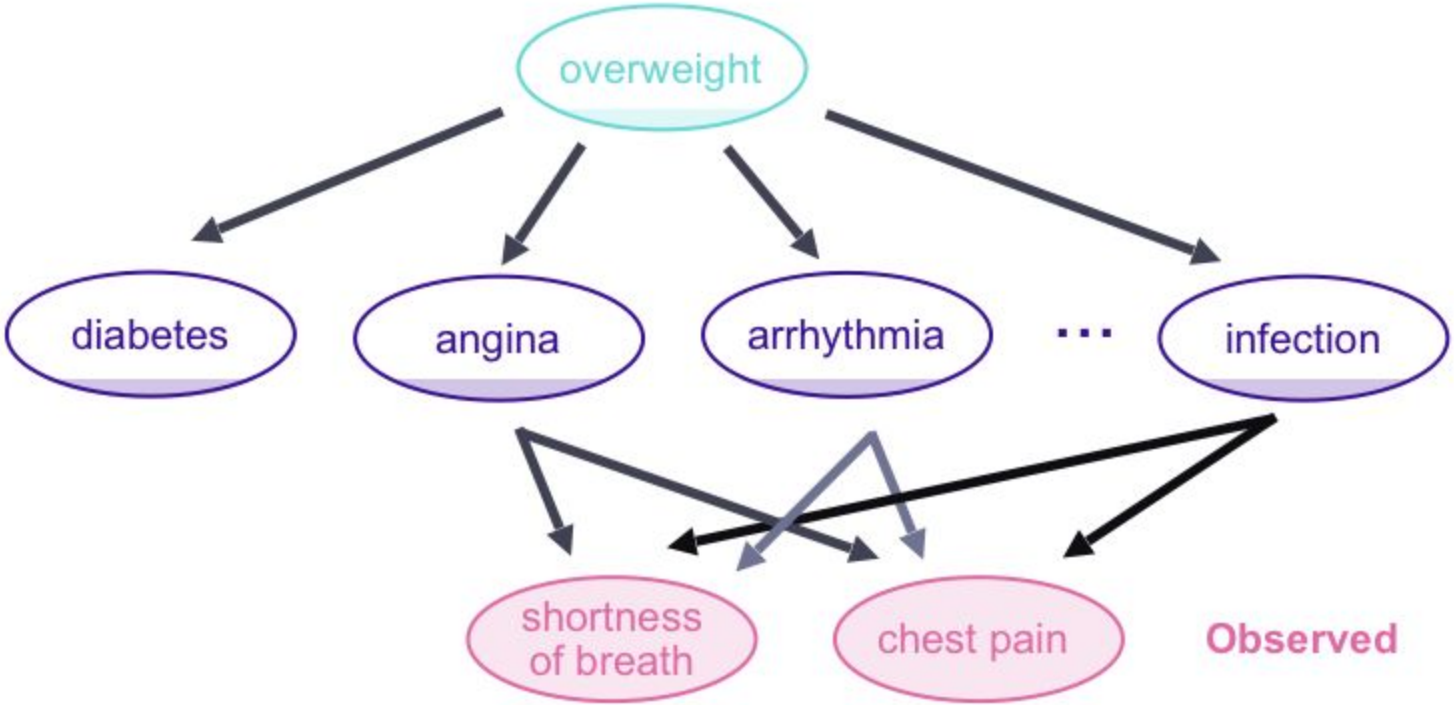
1. **Abduction**: update P(λ) to P(λ | S=T, C=T)

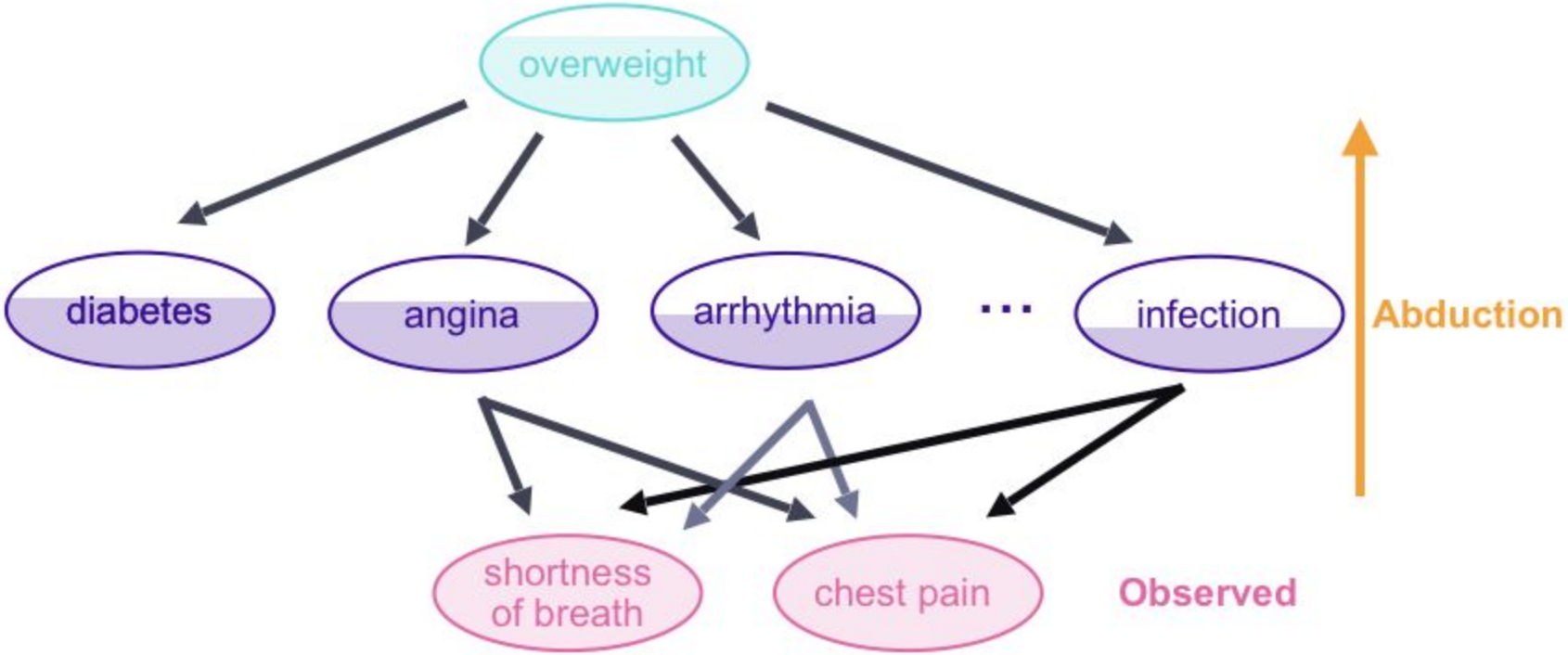
2. **Action**: Apply do(.) operator to force S=F

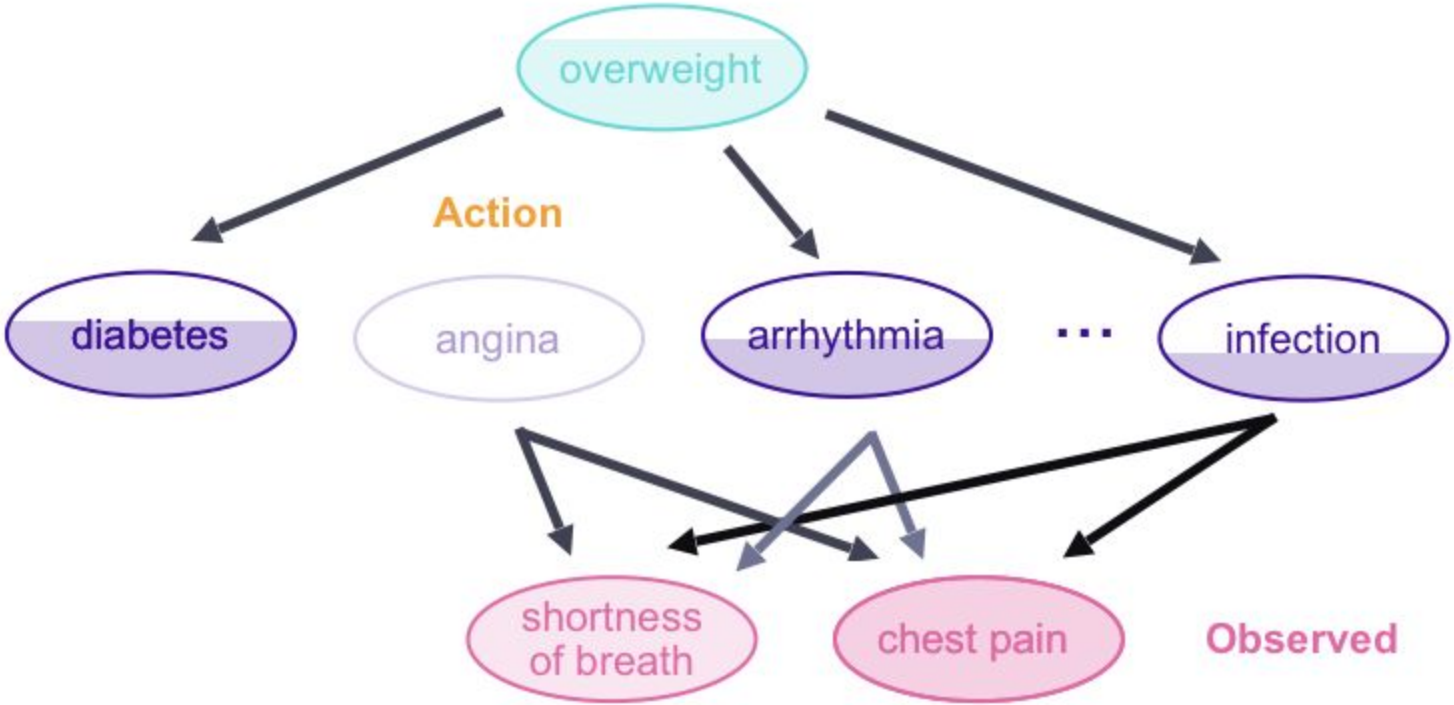
3. **Predict**: Compute P(C=F) in model with do(S=F) & P(λ | S=T, C=T)

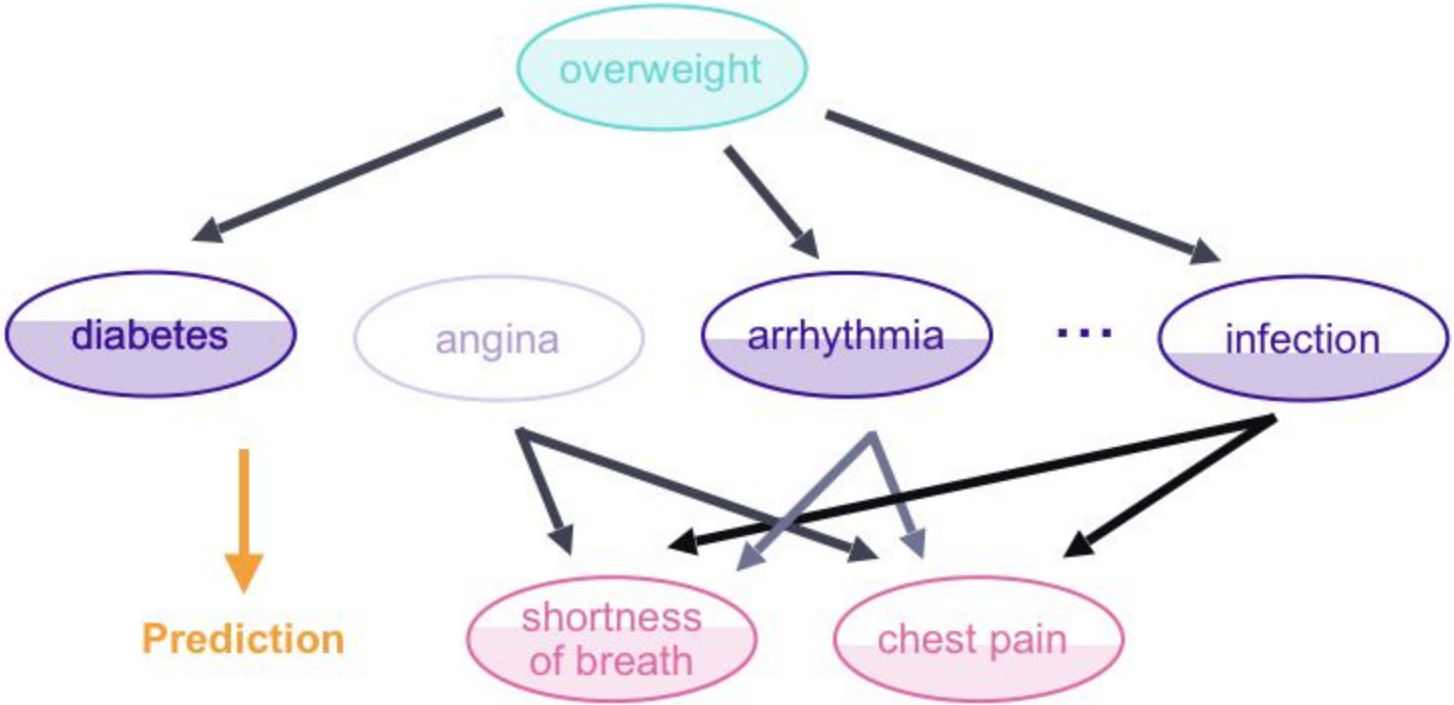
Example, compute: **P(Reduce Symptoms | Observe Symptoms, do(Cure Disease))**

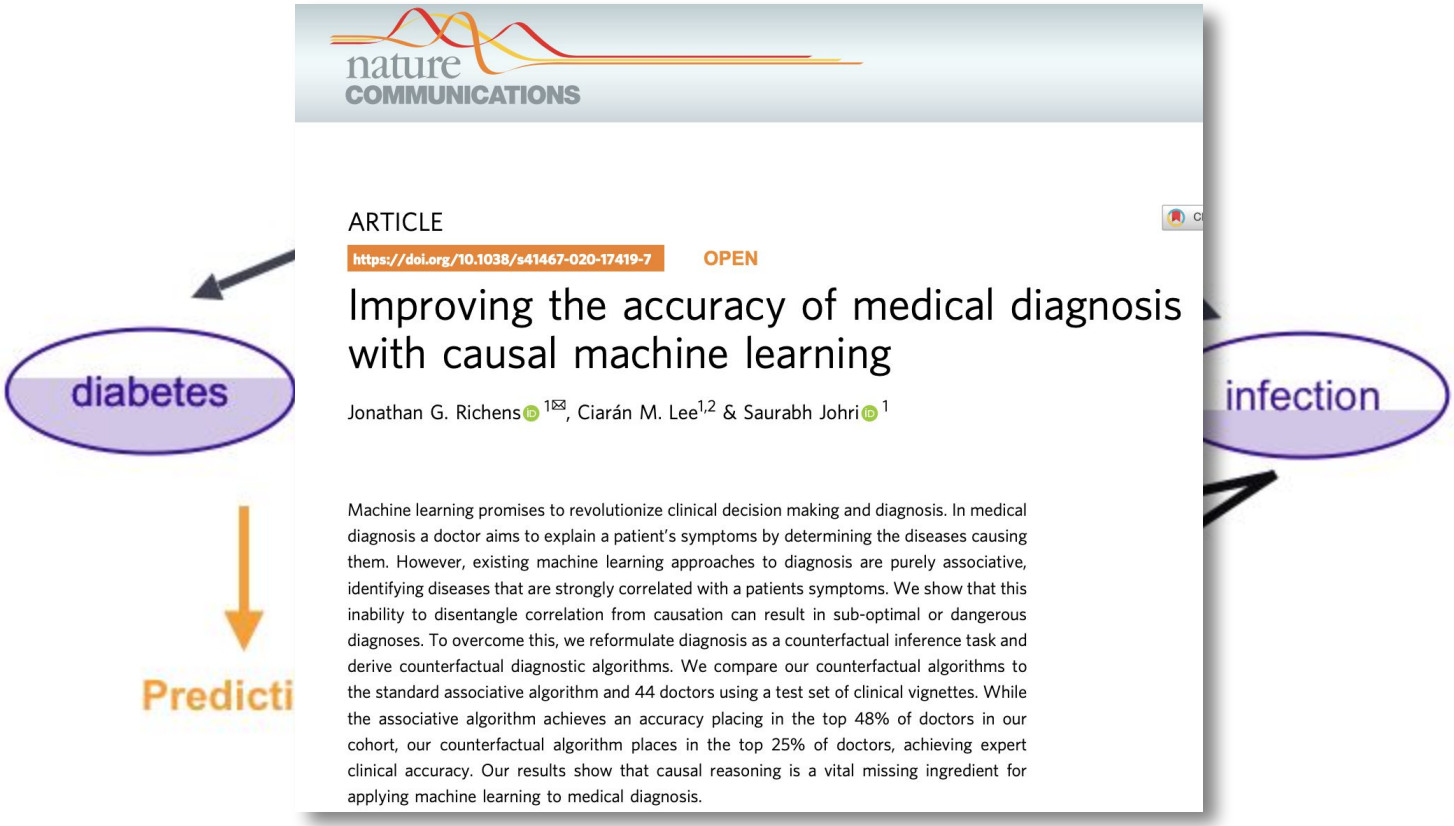



overweight
diabetes
angina
arrhythmia
...
infection
shortness of breath
chest pain
Observed


overweight
diabetes
angina
arrhythmia
...
infection
Abduction
shortness of breath
chest pain
Observed


overweight
Action
diabetes
angina
arrhythmia
...
infection
shortness of breath
chest pain
Observed


overweight
diabetes
angina
arrhythmia
...
infection
Prediction
shortness of breath
chest pain



nature COMMUNICATIONS

ARTICLE

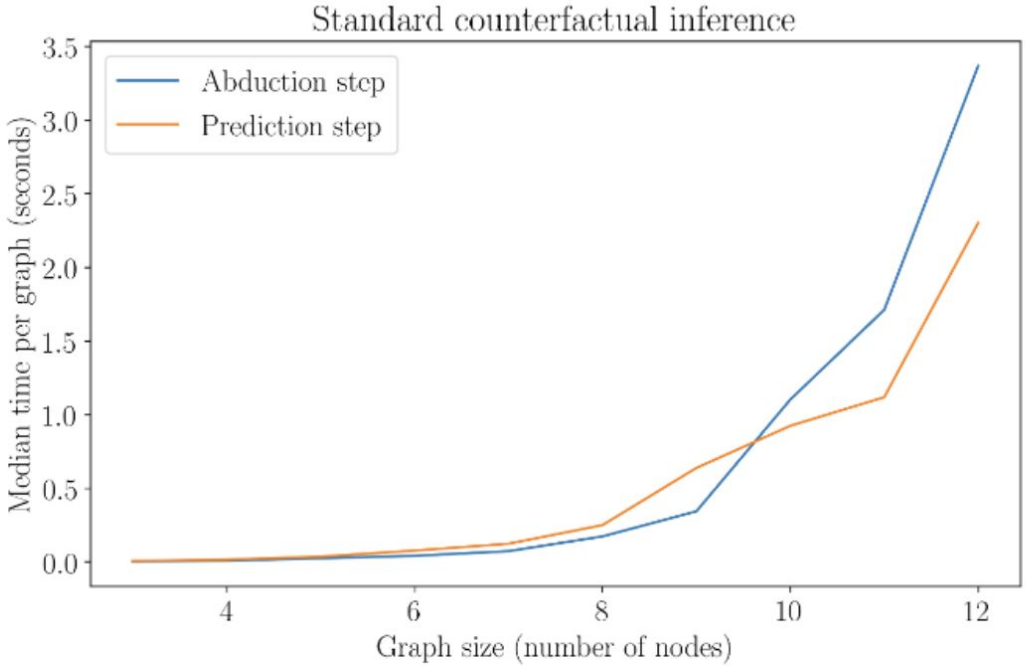
https://doi.org/10.1038/s41467-020-17419-7 OPEN

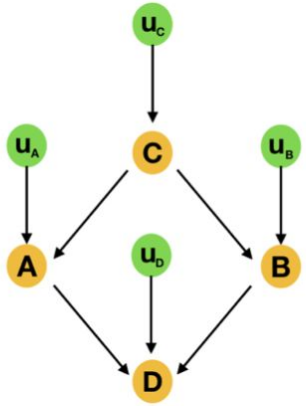
Improving the accuracy of medical diagnosis with causal machine learning

Jonathan G. Richens[1✉], Ciarán M. Lee[1,2] & Saurabh Johri[1]

Machine learning promises to revolutionize clinical decision making and diagnosis. In medical diagnosis a doctor aims to explain a patient's symptoms by determining the diseases causing them. However, existing machine learning approaches to diagnosis are purely associative, identifying diseases that are strongly correlated with a patients symptoms. We show that this inability to disentangle correlation from causation can result in sub-optimal or dangerous diagnoses. To overcome this, we reformulate diagnosis as a counterfactual inference task and derive counterfactual diagnostic algorithms. We compare our counterfactual algorithms to the standard associative algorithm and 44 doctors using a test set of clinical vignettes. While the associative algorithm achieves an accuracy placing in the top 48% of doctors in our cohort, our counterfactual algorithm places in the top 25% of doctors, achieving expert clinical accuracy. Our results show that causal reasoning is a vital missing ingredient for applying machine learning to medical diagnosis.

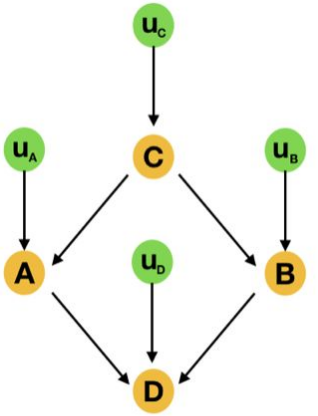
How efficient is counterfactual inference?

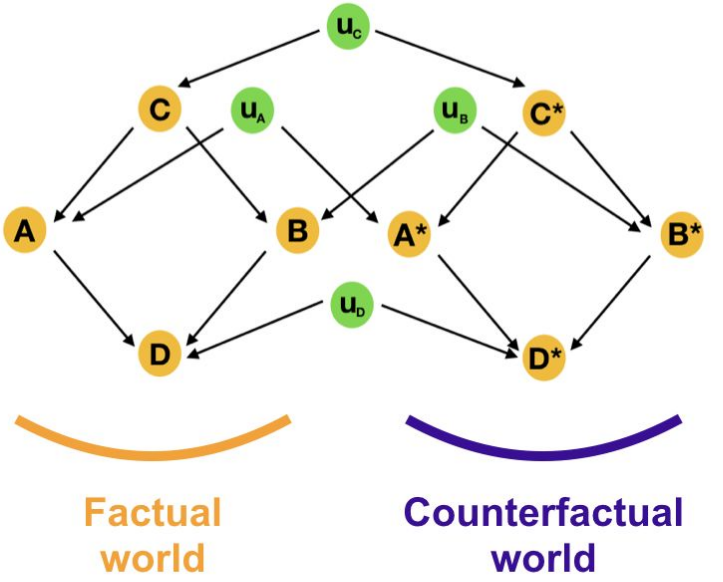




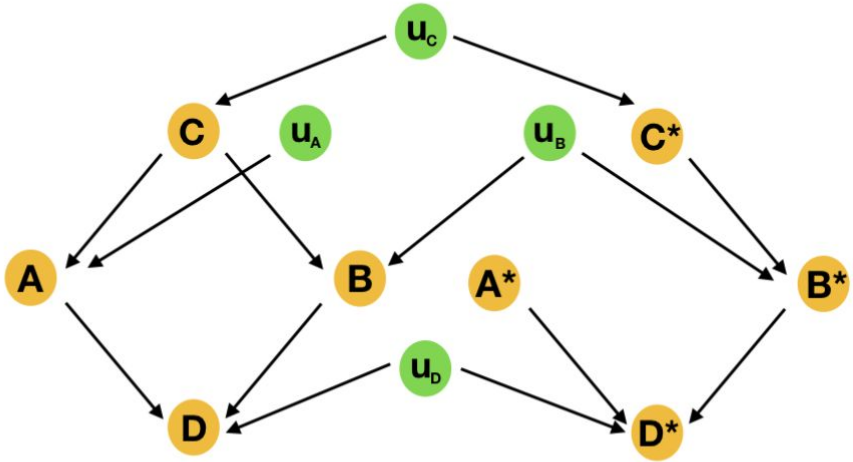
Abduction: update $P(u_A, u_B, u_C, u_D)$ given evidence

Efficient counterfactual inference with Twin Networks





Efficient counterfactual inference with Twin Networks

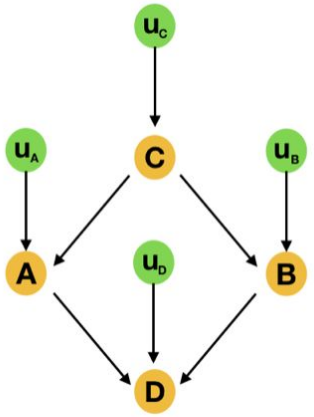


Compute counterfactual

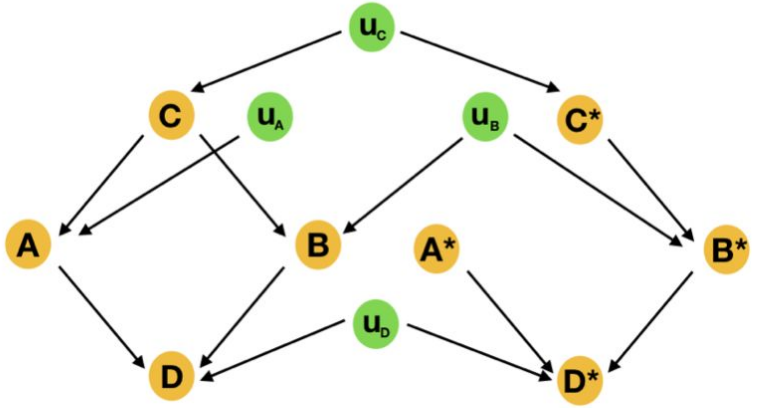
Standard: P(D | D=T, do(A=F))
1. Abduction
2. Action
3. Prediction

Twin: P(D* | D=T, A*=F)
Bayesian Inference on **Twin network**

Efficient counterfactual inference with Twin Networks

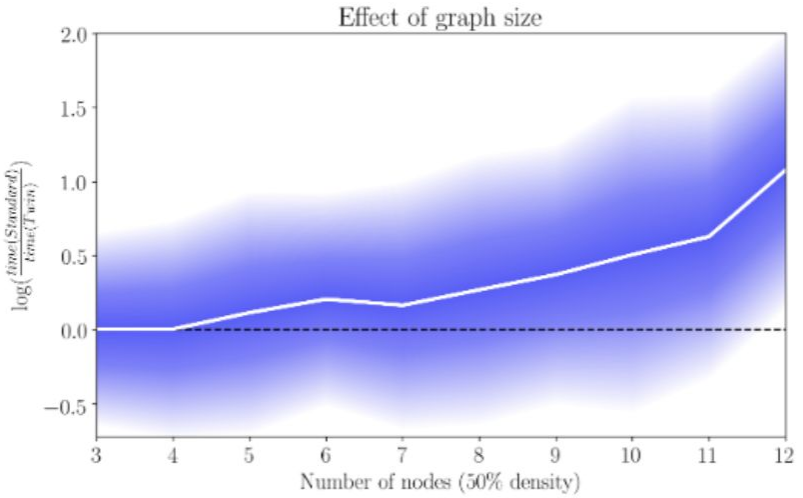


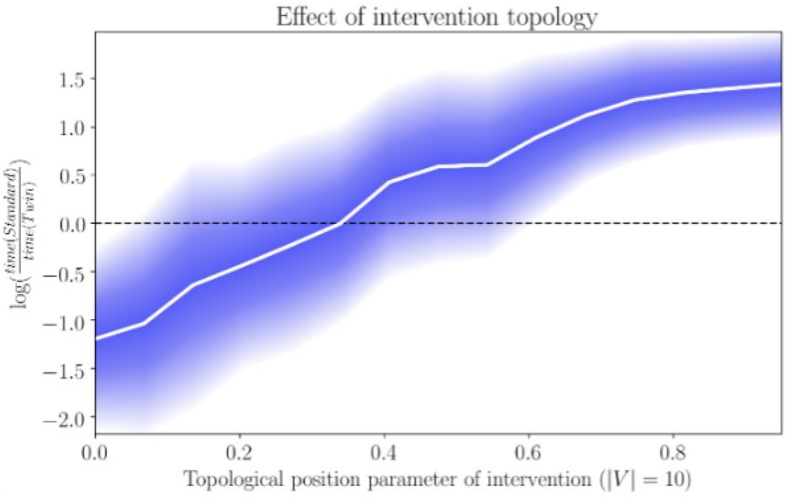
Update, make changes, & predict



Make changes in one, & predict

Computational advantage of Twin Networks



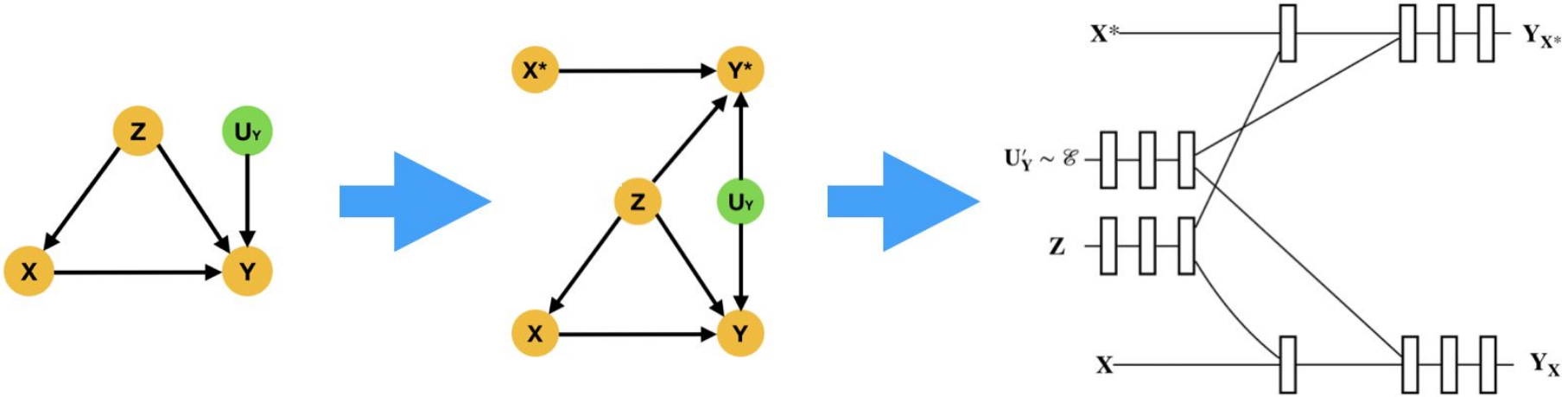


- Twin networks aren't a manifestly new way of doing counterfactual inference, but their graphical nature exposes ways to speed up the inference, either via parallelisation or by exploiting conditional independence relations

- Importantly, their graphical nature makes them very amenable to deep learning

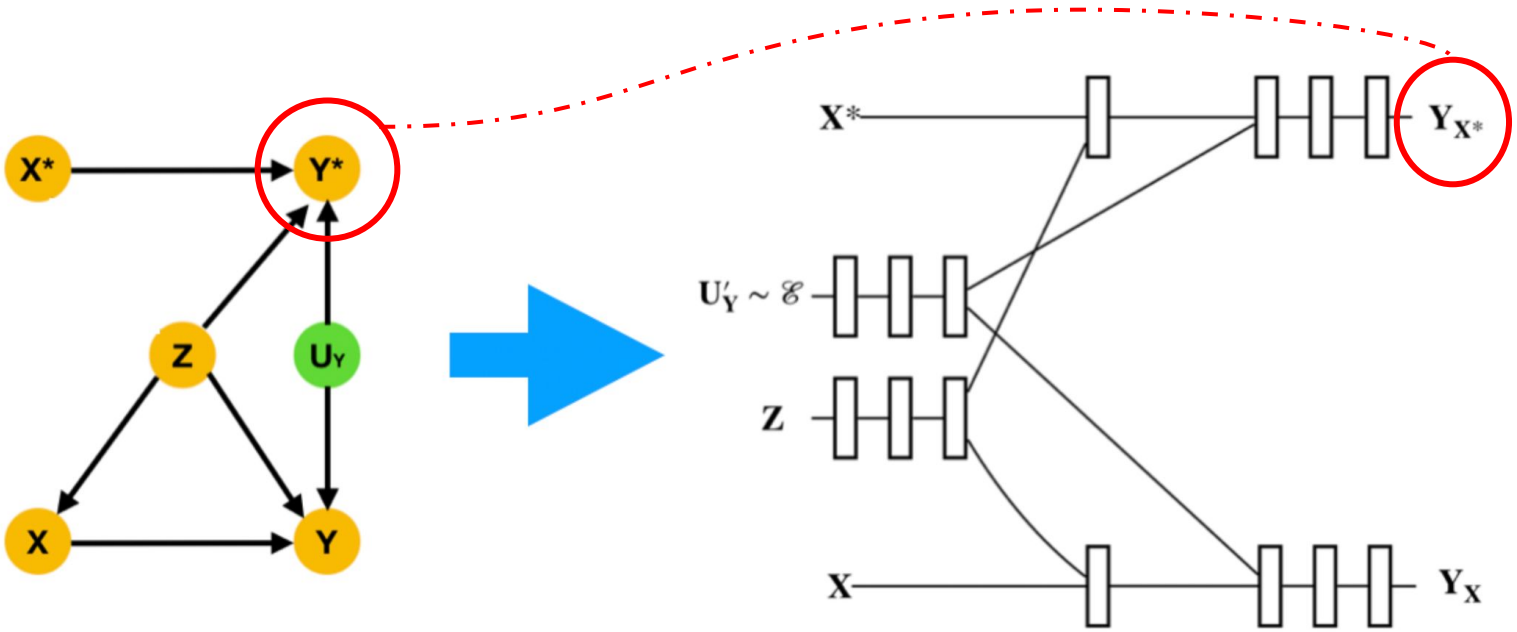
Learning problem II: Deep Twin Networks

Learning counterfactual distributions from 3rd layer



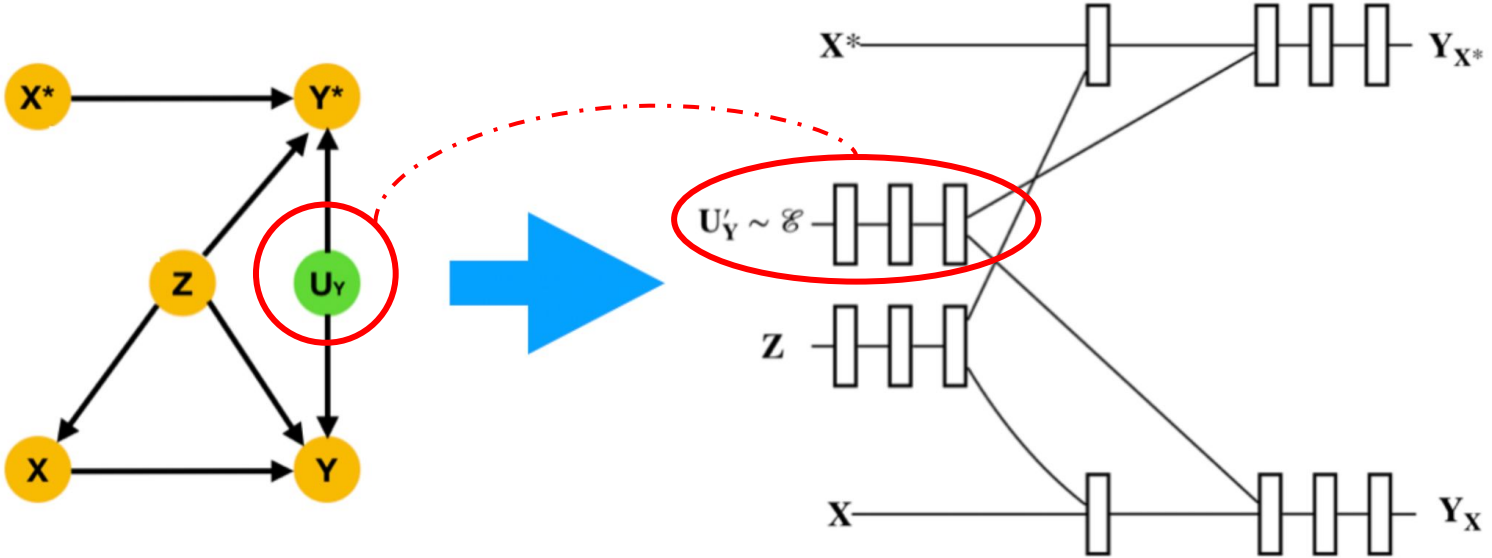
- Graphical nature of twin networks makes them very amenable to deep learning
- Yields simple neural network architectures that, when trained, yield full causal models that are capable of counterfactual inference

Training Deep Twin Networks



- What's the label of the red node? E(Y* | X*, Z) in Counterfactual network corresponds to E(Y | do(X), Z) in Factual network
- Estimate E(Y | do(X), Z) and use as Y* label, then train Deep Twin Net by minimising MSE on both heads

Training Deep Twin Networks

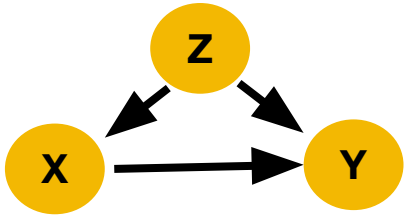


- Write Y=f(X, Z, U_Y) where U_Y~q(U_Y) as:

  Y=f(X, Z, g(U_Y')) where U_Y=g(U_Y') and U_Y'~N(0,1)

- Thus we want to learn functions f and g by sampling U_Y'~N(0,1) and passing through network with X, X* and Z.

Now, let's compute some counterfactuals!



Let's introduce some new notation:

$Y_{X=x}$ is the value of Y when we intervene with do(X=x)

We can now introduce some new causal questions of interest:

$P(Y_{X=0} = 0 \mid Y = 1, X=1, Z)$, for Y, X binary.

This is the **Probability of Necessity:** the probability event Y would not have occurred without event X occurring, given that X, Y did in fact occur in context Z

This will help us answer water spring problem

Are counterfactuals identifiable?

$$B = g(A, U),\ U \sim q(U)$$

$$= \begin{cases} A \text{ if } U = 0 \\ 0 \text{ if } U = 1 \\ 1 \text{ if } U = 2 \\ \text{NOT } A \text{ if } U = 3 \end{cases}$$

This model is completely specified by q(U):
That is, by 3 parameters

Are counterfactuals identifiable?

$$B = g(A, U),\ U \sim q(U)$$

$$= \begin{cases} A \text{ if } U = 0 \\ 0 \text{ if } U = 1 \\ 1 \text{ if } U = 2 \\ \text{NOT } A \text{ if } U = 3 \end{cases}$$

Observations/Interventions { P(B | A) } only restricts 2 parameters

$$P(B = 0 \,|\, A = 0) = q(U = 0) + q(U = 1),$$
$$P(B = 0 \,|\, A = 1) = q(U = 1) + q(U = 3)\,.$$

$$P(B_{A=0} = 0, B_{A=1} = 1) = q(U = 0)$$

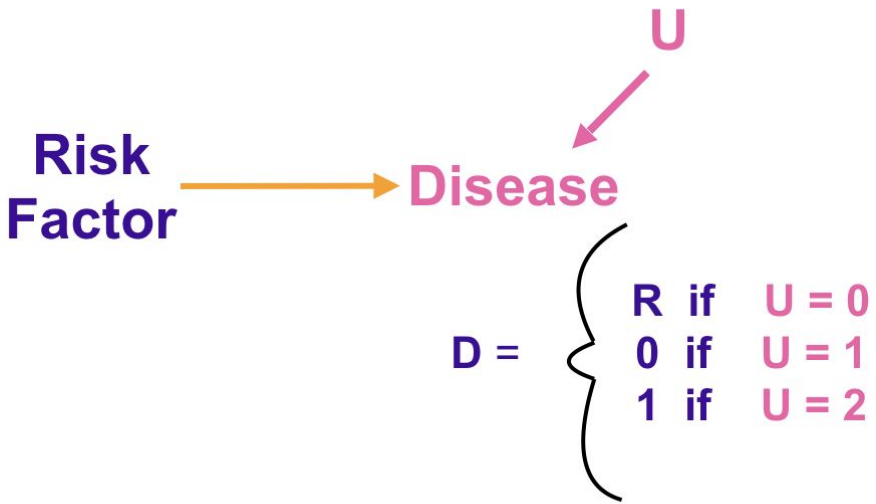
Can never use P(B|A) to learn q(U=0), hence counterfactual not identifiable even though we know all observations & interventions

Hence two models which agree on all observations and interventions can give different answers to the same counterfactual question

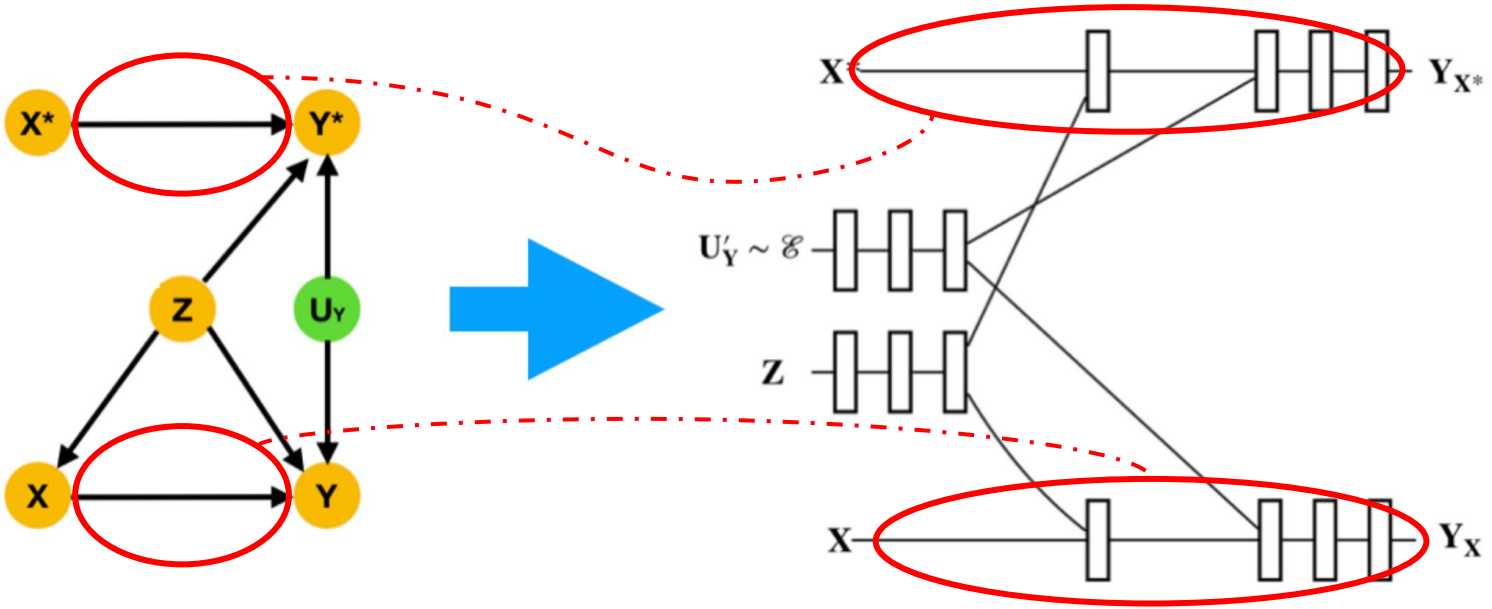
By constraining functions, can identify counterfactuals

Monotonicity says that presence of risk factor never decreases risk of disease:

"It's not possible to have risk factor and zero probability of developing disease"



Imposing identifiability constraints



- Enforce during training that functions learned by network satisfy identifiability. In case of binary variables this amounts to enforcing monotonicity
- In an updated version of our paper (coming to arXiv soon!) we explore **new constraints** on counterfactuals for **categorical** variables!

The full technical details are in our paper…

Algorithm 1: Training a deep twin network

Input: X: Treatment, Z: Confounders, X^*: Counterfactual Treatment; Y: Outcome; C: DAG of causal structure; I: loss imposing identifiability constraint

Output: F: trained deep twin network

Set F's architecture to match twin network representation of C, as in Figure 2
To obtain label for counterfactual head, first estimate $P(Z|X)$
Then, for x, z find $z' =$ with closest $P(z'|x)$ to $P(z|x)$
Set $y^* \leftarrow y(z', x^*)$, yielding training dataset $\mathcal{D} := \{X, X^*, Z; Y, Y^*\}$
for $x, x^*, z; y, y^* \in \mathcal{D}$ and $u_y \sim \mathcal{N}(0, 1)$ **do**
    $y', y^{'*} = F(x, x^*, u_y, z)$
    Train F by minimizing $MSE(y, y') + MSE(y^*, y^{'*}) + I(\mathcal{D})$
end for

Algorithm 2: Counterfactual Inference

Input: X: Treatment, U_Y: Noise, Z: Confounders, X^*: Counterfactual Treatment; Y: Outcome Y^*: Counterfactual Outcome; F : Trained deep twin network; Q: desired counterfactual query (in this example, $Y_{X=x'} = y' \mid X = x, Y = y, Z = z$)

Output: P(Q): Estimated distribution of Q.

Convert P(Q) to twin network distribution: $P(Y_{X=x'} = y' \mid X = x, Y = y, Z = z) \rightarrow P(Y^* = y' \mid X = x, Y = y, X^* = x')$
Compute $P(Y^* = y' \mid X = x, Y = y, X^* = x')$:
for $x, x', z \in \mathcal{D}_{test}$ **do**
    for $[u_y]_N \sim \mathcal{N}(0, 1), N \in \mathbb{N}$ **do**
        Sample $(\tilde{y}, \tilde{y}^* = F(x, x', u_y, z))$ such that $\tilde{y} = y$, for example using Rejection sampling, Importance sampling, etc.
        Frequency of these samples for which $\tilde{y}^* = y'$ yields P(Q)
    end for
end for

Should we protect water springs in this manner?

Method	P(N)	P(S)	P(N&S)
KW Median Child *Cuellar et al. 2020*	0.12 ± 0.01	-	-
KW TN Median Child	0.13598 ± 0.049	0.09811 ± 0.031	0.31778 ± 0.012
KW TN Test Set	0.06273 ± 0.020	0.03914 ± 0.016	0.08521 ± 0.034

$$P(\text{Necessity}) = P(\text{Diarrhoea}_{\text{Protect Springs=No}} = \text{No} \mid \text{Diarrhoea} = \text{Yes}, \text{Protect Springs=Yes}, Z)$$

$$P(\text{Sufficiency}) = P(\text{Diarrhoea}_{\text{Protect Springs=Yes}} = \text{Yes} \mid \text{Diarrhoea} = \text{No}, \text{Protect Springs=No}, Z)$$

$$P(\text{Nec. \& Suff.}) = P(\text{Diarrhoea}_{\text{Protect Springs=No}} = \text{No}, \text{Diarrhoea}_{\text{Protect Springs=Yes}} = \text{Yes} \mid Z)$$

- Exposure to water-based bacteria is not a necessary, sufficient, nor a necessary-and-sufficient condition to exhibit diarrhoea.
- This provides evidence that protecting water springs in this manner has little effect on the development of diarrhoea in children in these populations

Check out our paper to dive deeper arXiv:2109.01904

Estimating the probabilities of causation via deep monotonic twin networks

Athanasios Vlontzos[1*], **Bernhard Kainz** [1,2], **Ciarán M. Gilligan-Lee**[3]

[1] BioMedIA, Imperial College London
[2] FAU Erlangen-Nuremberg, [3] Spotify & University College London

Abstract

There has been much recent work using machine learning to answer causal queries. Most focus on interventional queries, such as the conditional average treatment effect. However, as noted by Pearl, interventional queries only form part of a larger hierarchy of causal queries, with *counterfactuals* sitting at the top. Despite this, our community has not fully succeeded in adapting machine learning tools to answer counterfactual queries. This work addresses this challenge by showing how to implement *twin network* counterfactual inference—an alternative to *abduction, action, & prediction* counterfactual inference—with deep learning to estimate counterfactual queries. We show how the graphical nature of twin networks makes them particularly amenable to deep learning, yielding simple neural network architectures that, when trained, are capable of counterfactual inference. Importantly, we show how to enforce known *identifiability* constraints during training, ensuring the answer to each counterfactual query is uniquely determined. We demonstrate our approach by using it to accurately estimate the probabilities of causation—important counterfactual queries that quantify the degree to which one event was a necessary or sufficient cause of another—on both synthetic and real data.

1 Introduction

Counterfactual queries establish if certain outcomes *would have* occurred had some precondition been different. Given evidence $\mathcal{E} = e$, counterfactual inference allows one to compute the probability a different outcome $\mathcal{E} = e'$ would have occurred—*counter-to-the-fact* $\mathcal{E} = e$—had some intervention taken place. The crucial difference between counterfactual and interventional queries is that the evidence the counterfactual query is "counter-to" can contain the variables we wish to intervene on or predict. An example counterfactual query is "Given I currently have a headache, would I not, had I taken medicine?". An interventional query is "What impact would medicine have on my headache?". The counterfactual query explicitly uses the evidence a headache is present, and asks whether medicine would have changed this fact. The interventional query asks what effect medicine would have on a headache for a given individual, but does not make use of the fact that a headache is currently present. Counterfactual inference has been applied to difficult problems in high profile sectors such as medicine (Richens, Lee, and Johri 2020; Oberst and Sontag 2019), legal analysis (Chockler and Halpern 2004; Lagnado, Gerstenberg, and Zultan 2013), fairness (Kusner et al. 2017; Kilbertus et al. 2017), explainability (Galhotra, Pradhan, and Salimi 2021), planning in reinforcement learning (Forney,

Check out our paper to dive deeper arXiv:2109.01904

Estimating the probabilities of causation via deep monotonic twin networks

Athanasios Vlontzos[1*], **Bernhard Kainz** [1,2], **Ciarán M. Gilligan-Lee**[3]

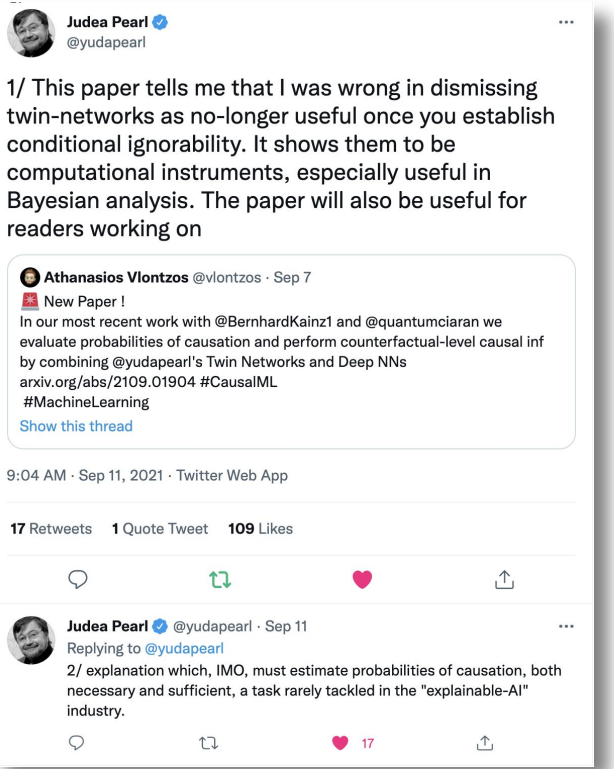
[1] BioMedIA, Imperial College London
[2] FAU Erlangen-Nuremberg, [3] Spotify & University College London

Abstract

There has been much recent work using machine learning to answer causal queries. Most focus on interventional queries, such as the conditional average treatment effect. However, as noted by Pearl, interventional queries only form part of a larger hierarchy of causal queries, with *counterfactuals* sitting at the top. Despite this, our community has not fully succeeded in adapting machine learning tools to answer counterfactual queries. This work addresses this challenge by showing how to implement *twin network* counterfactual inference—an alternative to *abduction, action, & prediction* counterfactual inference—with deep learning to estimate counterfactual queries. We show how the graphical nature of twin networks makes them particularly amenable to deep learning, yielding simple neural network architectures that, when trained, are capable of counterfactual inference. Importantly, we show how to enforce known *identifiability* constraints during training, ensuring the answer to each counterfactual query is uniquely determined. We demonstrate our approach by using it to accurately estimate the probabilities of causation—important counterfactual queries that quantify the degree to which one event was a necessary or sufficient cause of another—on both synthetic and real data.

1 Introduction

Counterfactual queries establish if certain outcomes *would have* occurred had some precondition been different. Given evidence $\mathcal{E} = e$, counterfactual inference allows one to compute the probability a different outcome $\mathcal{E} = e'$ would have occurred—*counter-to-the-fact* $\mathcal{E} = e$—had some intervention taken place. The crucial difference between counterfactual and interventional queries is that the evidence the counterfactual query is "counter-to" can contain the variables we wish to intervene on or predict. An example counterfactual query is "Given I currently have a headache, would I not, had I taken medicine?". An interventional query is "What impact would medicine have on my headache?". The counterfactual query explicitly uses the evidence a headache is present, and asks whether medicine would have changed this fact. The interventional query asks what effect medicine would have on a headache for a given individual, but does not make use of the fact that a headache is currently present. Counterfactual inference has been applied to difficult problems in high profile sectors such as medicine (Richens, Lee, and Johri 2020; Oberst and Sontag 2019), legal analysis (Chockler and Halpern 2004; Lagnado, Gerstenberg, and Zultan 2013), fairness (Kusner et al. 2017; Kilbertus et al. 2017), explainability (Galhotra, Pradhan, and Salimi 2021), planning in reinforcement learning (Forney,



Judea Pearl @yudapearl

1/ This paper tells me that I was wrong in dismissing twin-networks as no-longer useful once you establish conditional ignorability. It shows them to be computational instruments, especially useful in Bayesian analysis. The paper will also be useful for readers working on

> **Athanasios Vlontzos** @vlontzos · Sep 7
> New Paper !
> In our most recent work with @BernhardKainz1 and @quantumciaran we evaluate probabilities of causation and perform counterfactual-level causal inf by combining @yudapearl's Twin Networks and Deep NNs
> arxiv.org/abs/2109.01904 #CausalML
> #MachineLearning
> Show this thread

9:04 AM · Sep 11, 2021 · Twitter Web App

17 Retweets **1** Quote Tweet **109** Likes

Judea Pearl @yudapearl · Sep 11
Replying to @yudapearl
2/ explanation which, IMO, must estimate probabilities of causation, both necessary and sufficient, a task rarely tackled in the "explainable-AI" industry.

17

Examples of counterfactuals at Spotify

- **Which Playlists to update:** which playlists Z “need” to be updated?

$$P(Y_{X=\text{update}} = \text{engaged}, Y_{X=\text{no update}} = \text{not engaged} \mid Z)$$

- **New content to enjoy:** If user Z listened to specific content and enjoyed it, which other content would they also have enjoyed?

$$P(Y_{X=\text{new content}} = \text{engage} \mid Y = \text{engage}, X = \text{current content}, Z)$$

And many more….

Conclusion

- Many interesting causal inference problems need to be solved to address important problems in industry and beyond, such as disentangling multiple treatments
- Range of *counterfactual* questions that yield invaluable insights beyond just average, or even heterogeneous, treatment effects
- Deep twin networks provide simple way to use ML to learn causal models and answer counterfactual questions
- Many more causal inference applications at Spotify beyond what we've discussed today
- We're hiring, so get in touch if you're interested!

ciaranl@spotify, or
ciaran.lee@ucl.ac.uk

@quantumciaran