Code: https://github.com/microsoft/dowhy
Paper: https://arxiv.org/abs/2011.04216

DoWhy: An End-to-End Library for Causal Inference

Amit Sharma (@amt_shrma), Emre Kıcıman (@emrek)
Microsoft Research

Causal ML group @ Microsoft:
https://www.microsoft.com/en-us/research/group/causal-inference/

From prediction to decision-making




Decision-making: Acting/intervening based on analysis

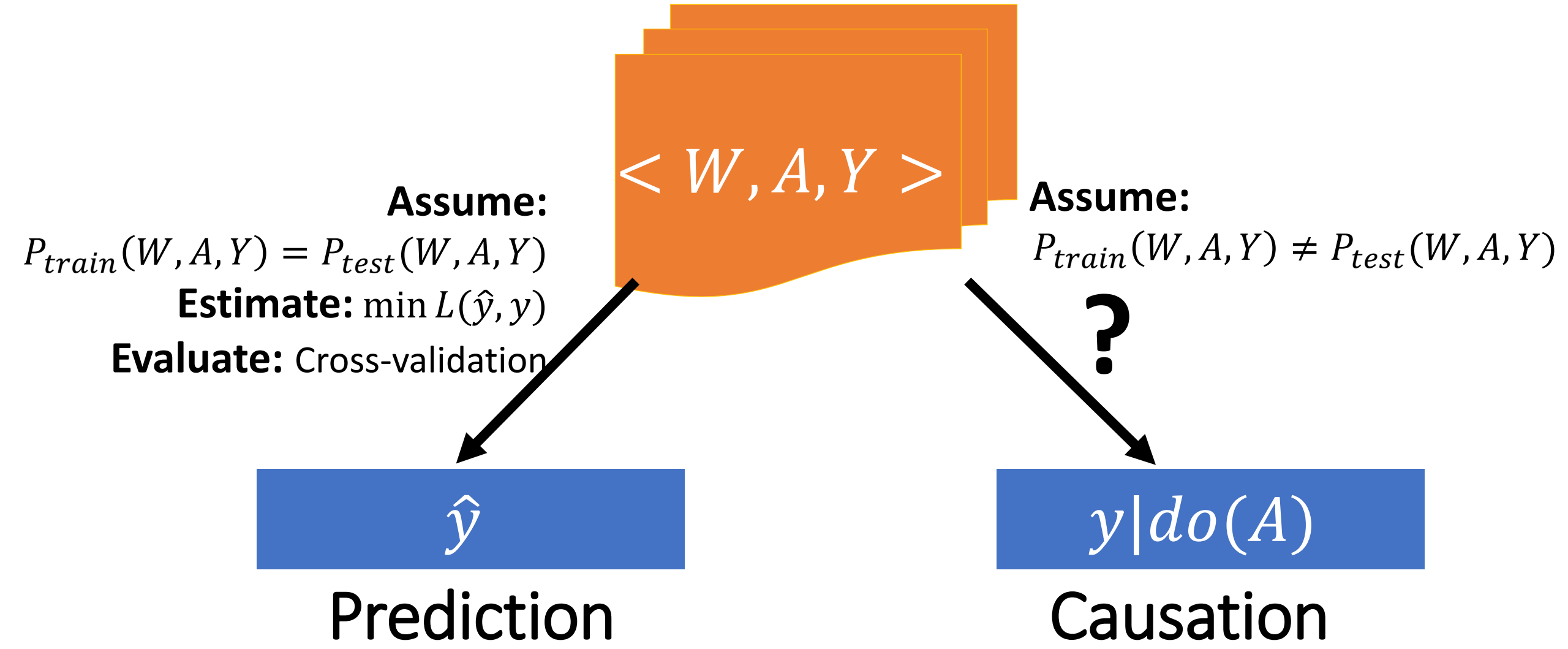
- Interventions break correlations used by conventional ML
- The feature with the highest importance score in a prediction model,
  - Need not be the best feature to act on
  - May not even affect the outcome at all!

For decision-making, need to find the features that **cause the outcome** & *estimate how the outcome would change if the features are changed.*

Many data science Qs are causal Qs

- **A/B experiments:** If I change the algorithm, will it lead to a higher success rate?
- **Policy decisions:** If we adopt this treatment/policy, will it lead to a healthier patient/more revenue/etc.?
- **Policy evaluation:** knowing what I know now, did my policy help or hurt?
- **Credit attribution:** are people buying because of the recommendation algorithm? Would they have bought anyway?

In fact, causal questions form the basis of almost all scientific inquiry



Hofman, **Sharma**, and Watts (2017). *Science*, 355.6324

Two Fundamental Challenges for Causal Inference

Multiple causal mechanisms and estimates can fit the same data distribution.

➡ **1. Assumptions**

Estimation about different data distributions than the training distribution (no easy “cross-validation”).

➡ **2. Evaluation**

Real World: **do(A=1)**

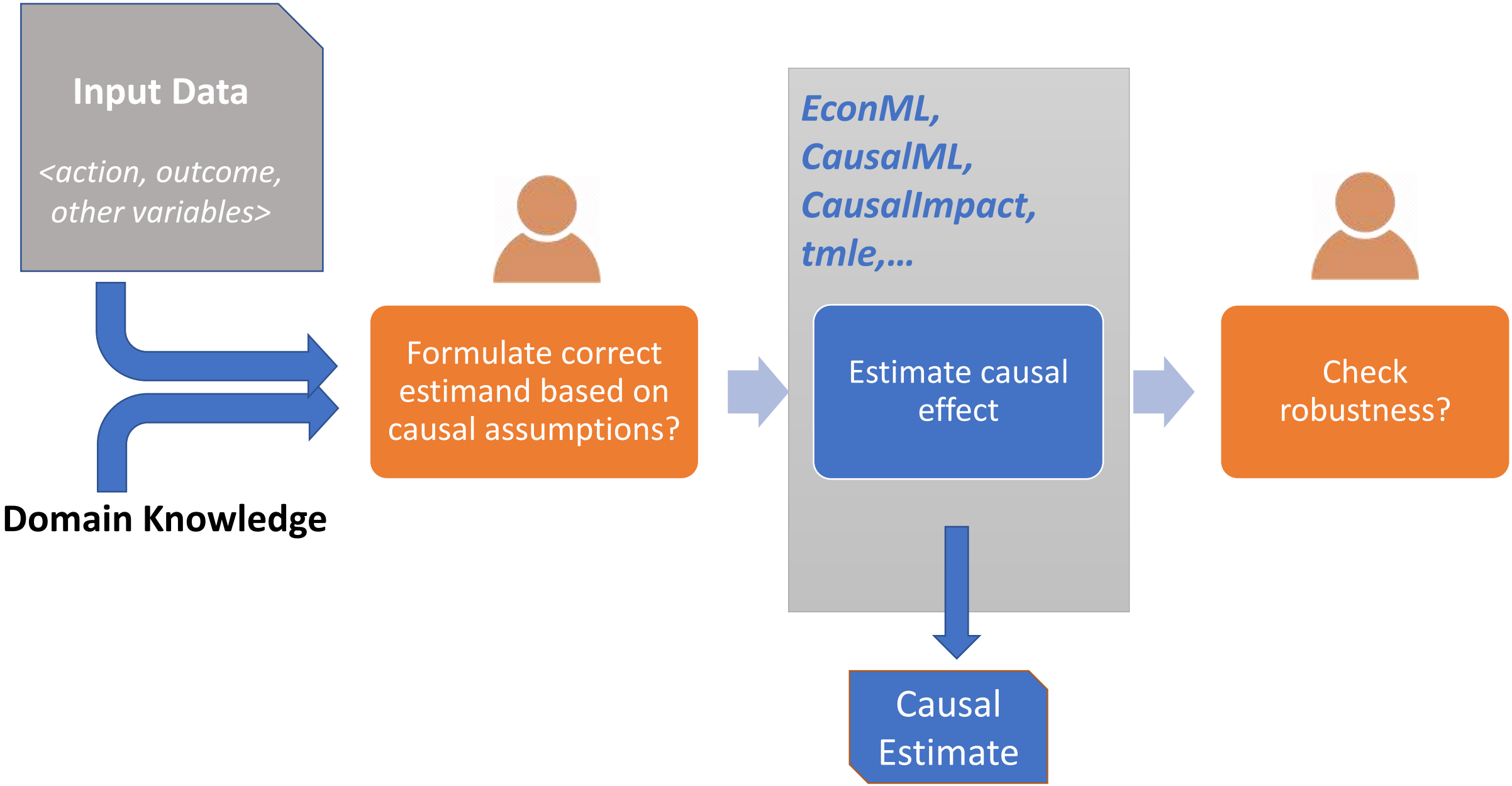
Counterfactual World: **do(A=0)**

We built DoWhy library to make assumptions front-and-center of any causal analysis.

- Transparent declaration of assumptions
- Evaluation of those assumptions, to the extent possible

Most popular causal library on GitHub (>2.4k stars, 300+ forks)
Taught in third-party tutorials and courses: O'Reilly, PyData, Northeastern, …

An end-to-end platform for doing causal inference

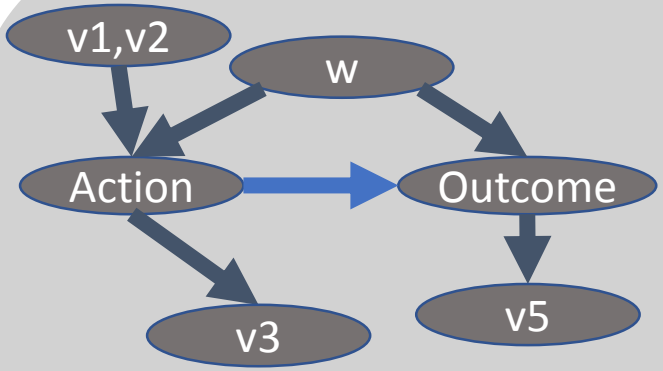

Input Data
<action, outcome, other variables>
Domain Knowledge
Formulate correct estimand based on causal assumptions?
EconML, CausalML, CausalImpact, tmle,…
Estimate causal effect
Causal Estimate
Check robustness?

DoWhy

Input Data

<action, outcome, other variables>

Domain Knowledge



Model causal mechanisms

- Construct a causal graph based on domain knowledge

Identify the target estimand

- Formulate correct estimand based on the causal model

Estimate causal effect

- Use a suitable method to estimate effect

Refute estimate

- Check robustness of estimate to assumption violations

Causal effect

DoWhy provides a general API for the four steps of causal inference

1. **Modeling:** Create a causal graph to encode assumptions.
2. **Identification:** Formulate what to estimate.
3. **Estimation:** Compute the estimate.
4. **Refutation:** Validate the assumptions.

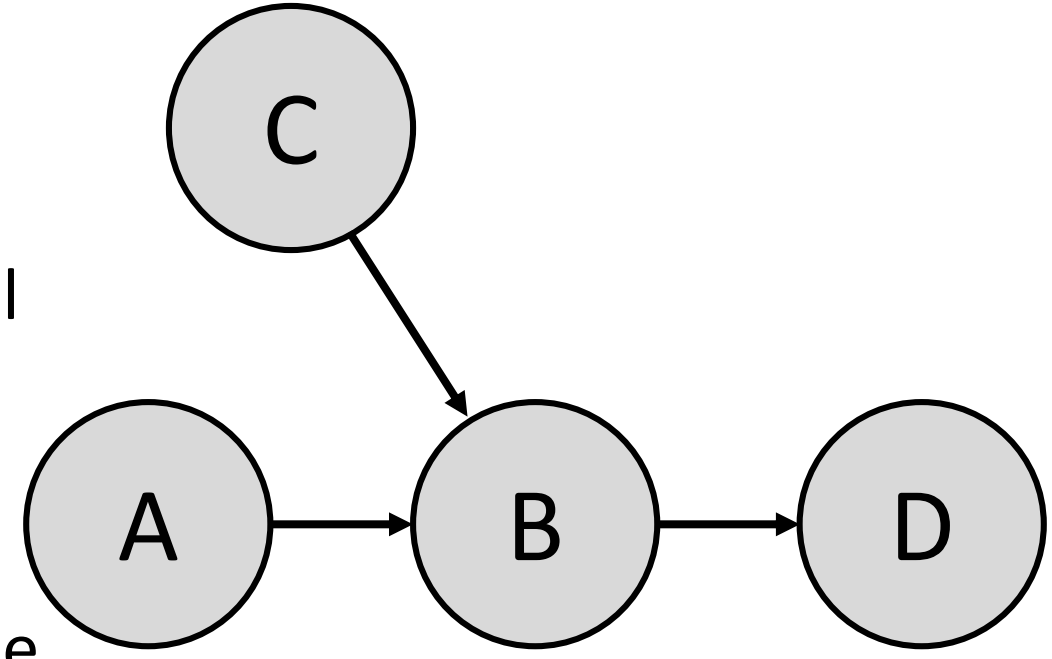
We'll discuss the four steps and show a code example using DoWhy.

I. Model the assumptions using a causal graph

Convert domain knowledge to a formal model of causal assumptions

- $A \rightarrow B$ or $B \rightarrow A$?

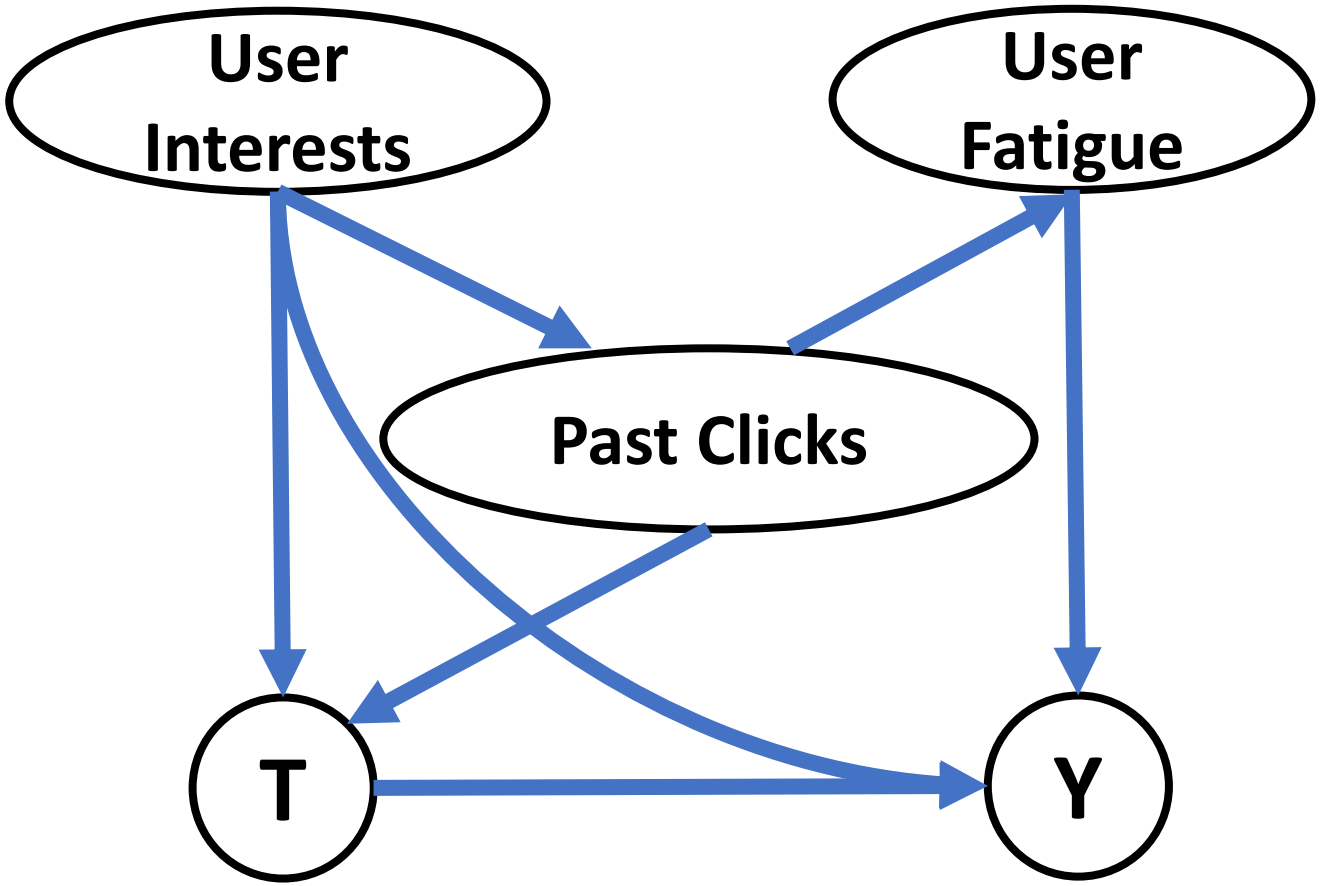
- Causal graph implies conditional statistical independences
  - E.g., $A \perp\!\!\!\perp C$, $D \perp\!\!\!\perp \mathrm{A} \mid \mathrm{B}$, …
  - Identified by *d-separation* rules [Pearl 2009]
- These assumptions significantly impact the causal estimate we'll obtain.



Key intuitions about causal graphs

- Assumptions are encoded by *missing edges*, and *direction* of edges
- Relationships represent stable and independent mechanisms
- Graph cannot be learnt from data alone
- Graphs are a tool to help us reason about a specific problem

Example Graph



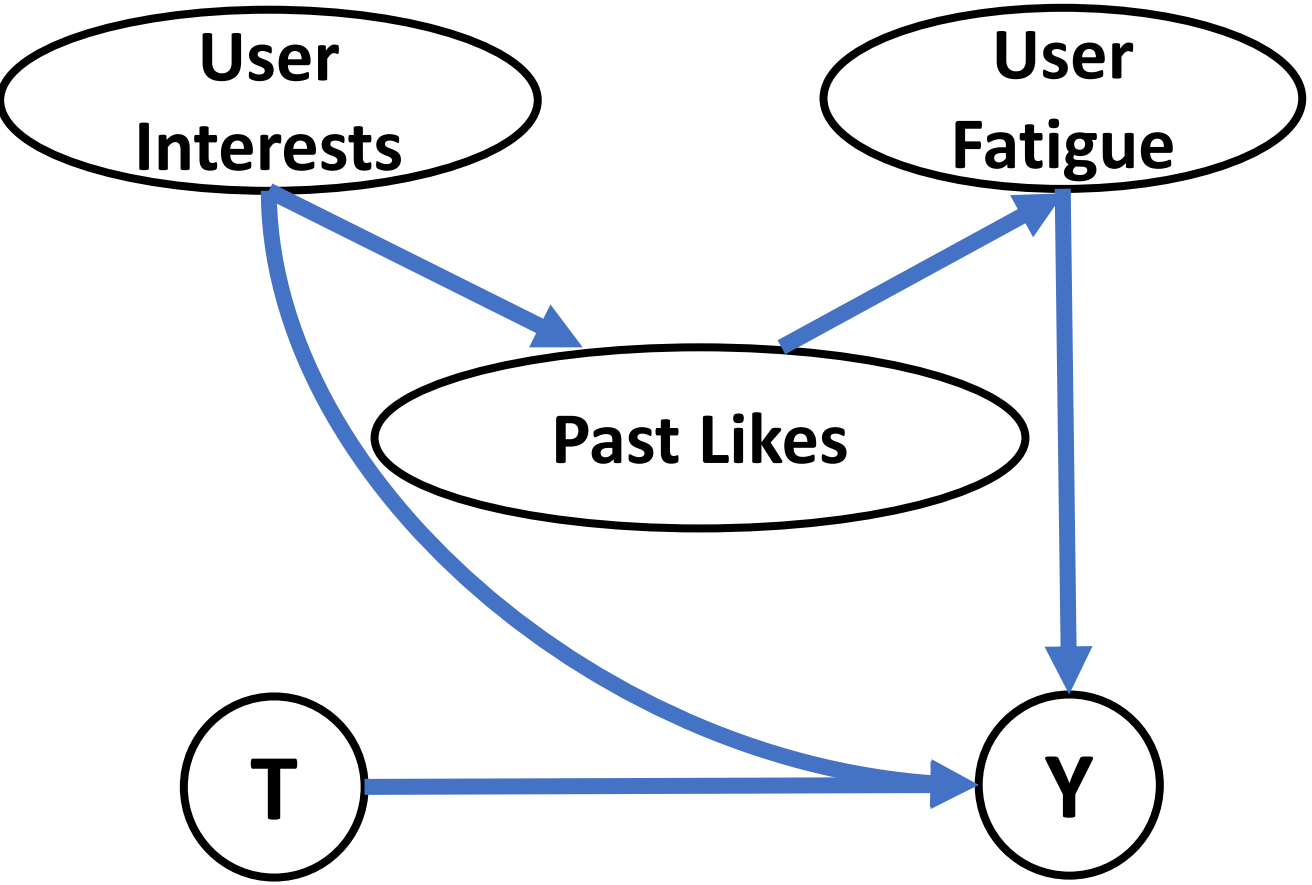
Assumption 1: User fatigue does not affect user interests

Assumption 2: Past clicks do not directly affect outcome

Assumption 3: Treatment does not affect user fatigue.

..and so on.

Intervention is represented by a new graph



Intervention graph:

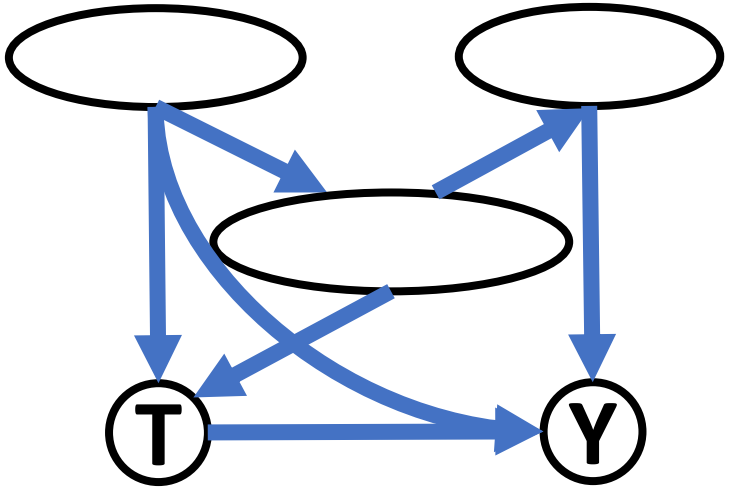
All edges to Treatment T removed, *keeping everything else the same*.

Represents new data distribution, referred as $do(T)$

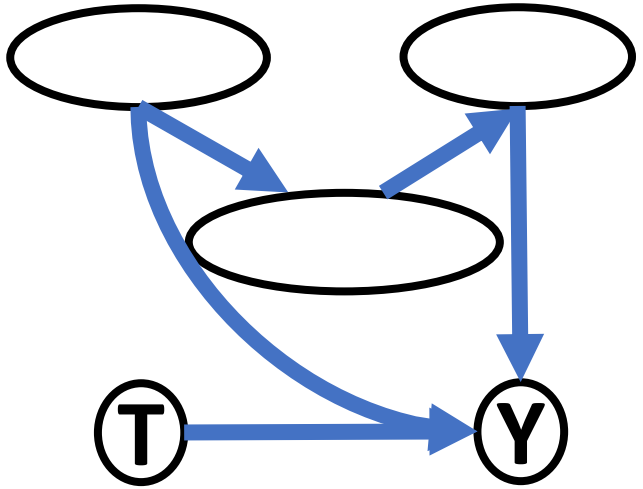
Causal effect: $P(Y|do(T))$

II. **Identification:** Formulate desired quantity and check if it is estimable from given data

Observed data generated by this graph



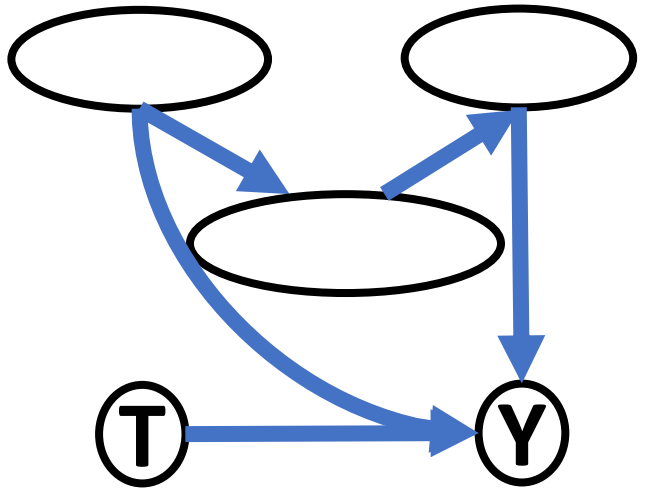
Want to answer questions about data that *will* be generated by intervention graph



How to represent quantities from right hand graph (e.g., $P(Y|do(T))$ using only statistical observations from data generated from left hand graph?

Trivial Example: Randomized Experiments

- Observed graph is same as intervention graph in randomized experiment!
  - Treatment T is already generated independent of all other features
  - ➔ $P(Y|do(T)) = P(Y|T)$
- **Intuition:** Generalize by simulating randomized experiment
  - When treatment T is caused by other features, Z, adjust for their influence to simulate a randomized experiment



Adjustment Formula and Adjustment Sets

Adjustment formula

$$p(Y|do(T)) = \sum_{Z} p(Y|T,Z)p(Z)$$

Where Z must be a valid adjustment set:

- The set of all parents of T
- Features identified via *backdoor criterion*
- Features identified via “towards necessity” criterion

Intuitions:

- The union of all features is *not* necessarily a valid adjustment set
- Why not always use parents? Sometimes parent features are unobserved

Many kinds of identification methods

Graphical constraint-based methods

- Randomized and natural experiments
- Adjustment Sets
  - Backdoor, “towards necessity”
- Front-door criterion
- Mediation formula

Identification under additional non-graphical constraints

- Instrumental variables
- Regression discontinuity
- Difference-in-differences

Many of these methods can be used through DoWhy.

III. **Estimation:** Compute the causal effect

Estimation uses observed data to compute the target probability expression from the Identification step.

For common identification strategies using adjustment sets,

$$E[Y|do(T=t), W=w] = E[Y|T=t, W=w]$$

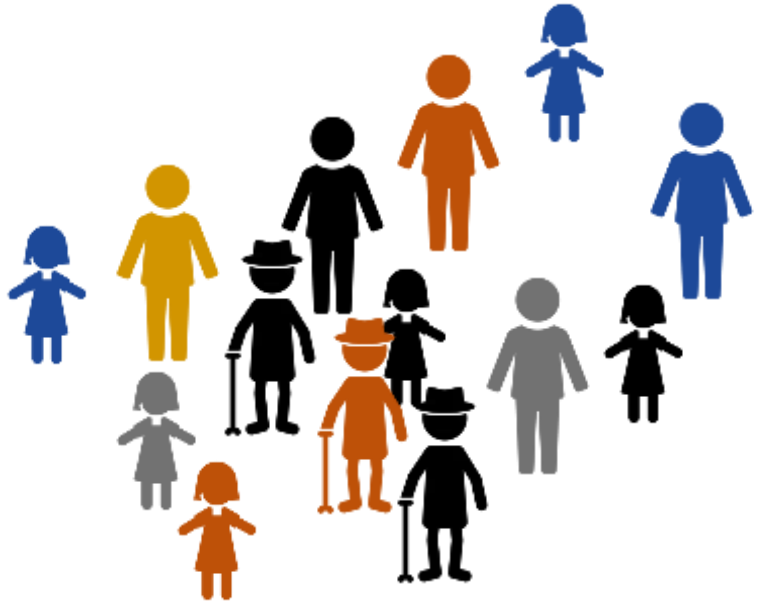
assuming W is a valid adjustment set.

- For binary treatment,

$$\text{Causal Effect} = E[Y|T=1, W=w] - E[Y|T=0, W=w]$$

Goal: Estimating conditional probability Y|T=t when all confounders W are kept constant.

Simple Matching: Match data points with the same confounders and then compare their outcomes



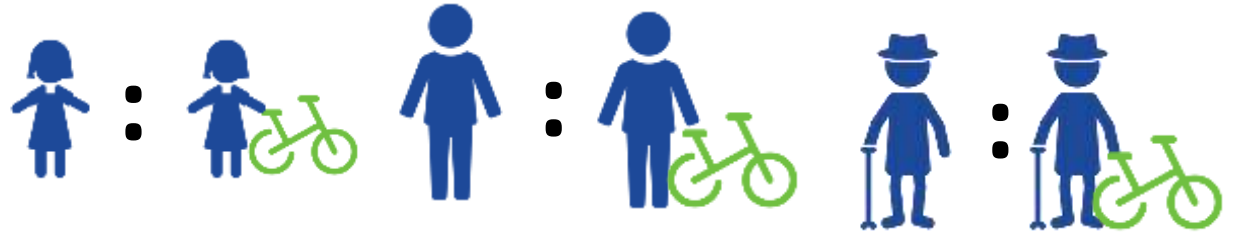
Control

Treatment (Cycling)

Simple Matching: Match data points with the same confounders and then compare their outcomes

Identify pairs of treated (j) and untreated individuals (k) who are similar or identical to each other.

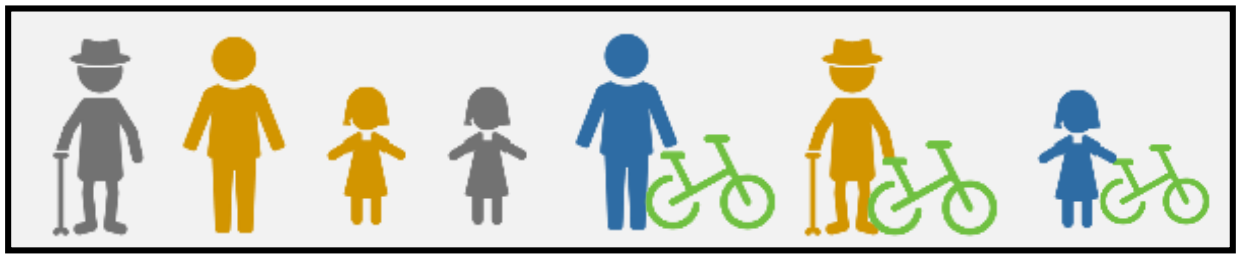
Match := $Distance(W_j, W_k) < \epsilon$



- Paired individuals have almost the same confounders.

  Causal Effect =

  $$\sum_{(j,k)\in Match}(y_j - y_k)$$



Challenges of building a good estimator

- **Variance:** If we have a stringent matching criterion, we may obtain very few matches and the estimate will be unreliable.
- **Bias:** If we relax the matching criterion, we obtain many more matches but now the estimate does not capture the target estimand.
- **Uneven treatment assignment:** If very few people have treatment, leads to both high bias and variance.

Need better methods to navigate the bias-variance tradeoff.

Machine learning methods can help find a better match for each data point

Synthetic Control: If a good match does not exist for a data point, can we create it synthetically?

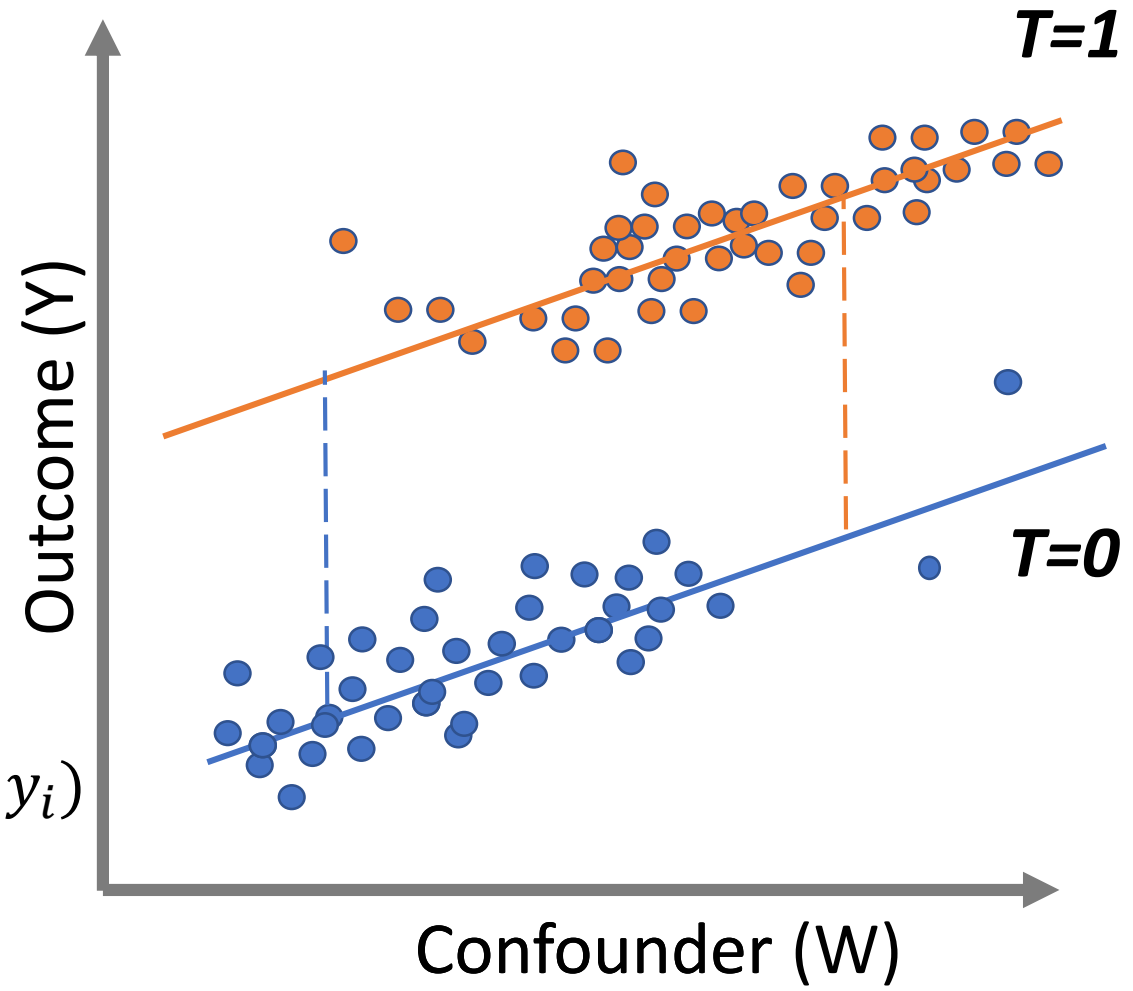
Learn $y = f_{t=0}(w)$,

$$y = f_{t=1}(w)$$

Assuming f approximates the true relationship between Y and W,

Causal Effect =

$$\sum_i t_i(y_i - f_{t=0}(w_i)) + (1 - t_i)(f_{t=1}(w_i) - y_i)$$



Other ML methods generalize estimation to continuous treatments

The standard predictor, $y = f(t, w) + \epsilon$ may not provide the right estimate for $\frac{\partial y}{\partial t}|$.

Double-ML [Chernozhukov et al. 2016]**:**

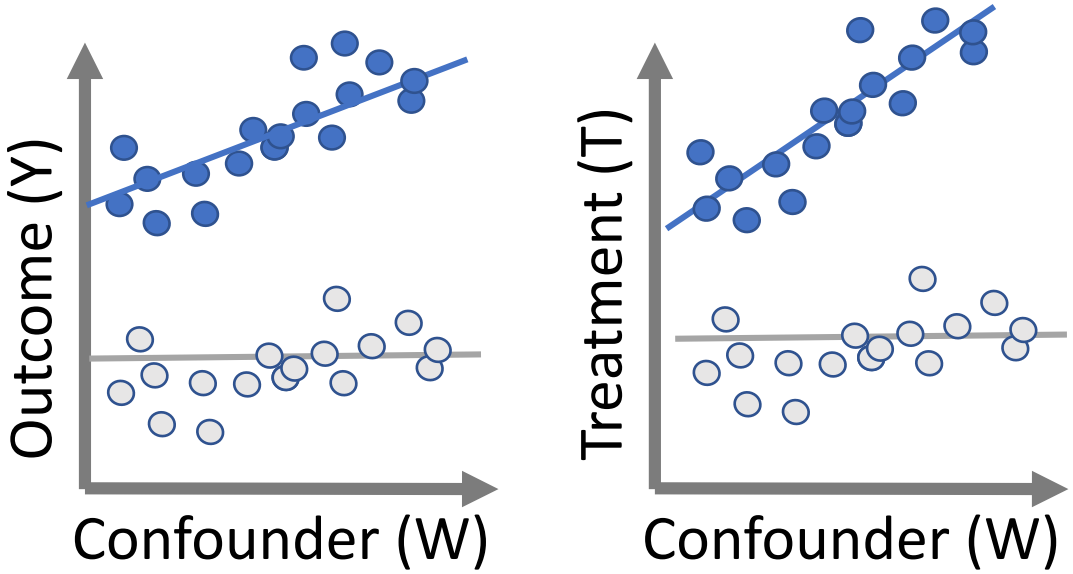
- Stage 1: Break down conditional estimation into two prediction sub-tasks.

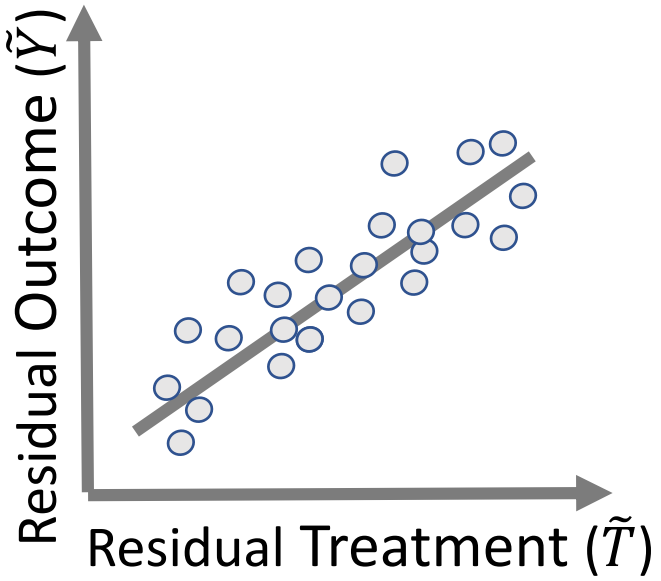
$$\hat{y} = g(w) + \tilde{y}$$
$$\hat{t} = h(w) + \tilde{t}$$

$\tilde{y}$ and $\tilde{t}$ refer to the unconfounded variation in Y and T respectively after conditioning on w.

- Stage 2: A final regression of $\tilde{y}$ on $\tilde{t}$ gives the causal effect.

$$\tilde{y} \sim \beta \tilde{t} + \epsilon$$





Depending on the dataset properties, different estimation methods can be used

Simple Conditioning

- Matching
- Stratification

Propensity Score-Based [Rubin 1983]

- Propensity Matching
- Inverse Propensity Weighting

Synthetic Control [Abadie et al.]

Outcome-based

- Double ML [Chernozhukov et al. 2016]
- T-learner
- X-learner [Kunzel et al. 2017]

Loss-Based

- R-learner [Nie & Wager 2017]

Threshold-based

- Difference-in-differences

All these methods can be called through DoWhy.
(directly or through the Microsoft EconML library)

IV. Robustness Checks: Test robustness of obtained estimate to violation of assumptions

Obtained estimate depends on many (untestable) assumptions.

Model:

Did we miss any unobserved variables in the assumed graph?

Did we miss any edge between two variables in the assumed graph?

Identify:

Did we make any parametric assumption for deriving the estimand?

Estimate:

Is the assumed functional form sufficient for capturing the variation in data?

Do the estimator assumptions lead to high variance?

Best practice: Do refutation/robustness tests for as many assumptions as possible

UNIT TESTS

Model:

- Conditional Independence Test

Identify:

- D-separation Test

Estimate:

- Bootstrap Refuter
- Data Subset Refuter

INTEGRATION TESTS

Test all steps at once.

- Placebo Treatment Refuter
- Dummy Outcome Refuter
- Random Common Cause Refuter
- Sensitivity Analysis
- Simulated Outcome Refuter /Synth-validation [Schuler et al. 2017]

All these refutation methods are implemented in DoWhy.
Caveat: They can refute a given analysis, *but cannot prove its correctness.*

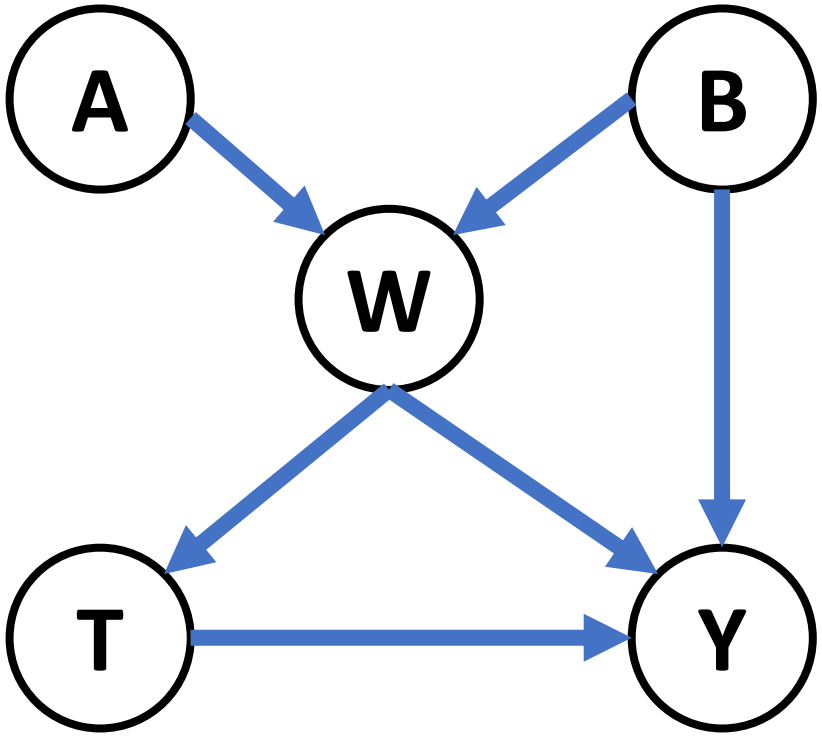
Example 1: Conditional Independence Refuter

Through its edges, each causal graph implies certain conditional independence constraints on its nodes. *[d-separation, Pearl 2009]*

Model refutation: Check if the observed data satisfies the assumed model's independence constraints.

- Use an appropriate statistical test for independence *[Heinze-Demel et al. 2018].*

- If not, the model is incorrect.



Conditional Independencies:

$A \perp\!\!\!\perp B$ $A \perp\!\!\!\perp \mathrm{T} \mid \mathrm{W}$ $B \perp\!\!\!\perp \mathrm{T} \mid \mathrm{W}$

Example 1: Placebo Treatment ("A/A") Refuter

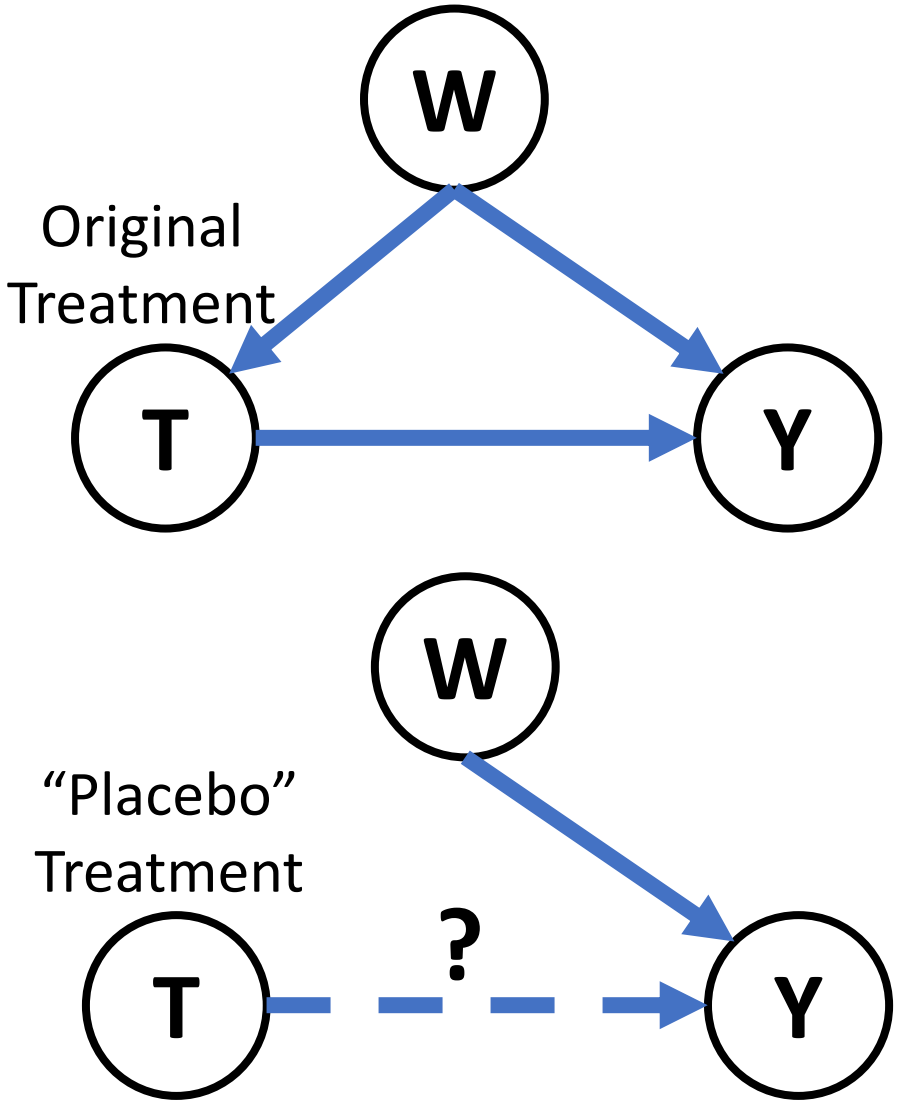
Q: *What if we can generate a dataset where the treatment does not cause the outcome?*

Then a correct causal inference method should return an estimate of zero.

Placebo Treatment Refuter:

Replace treatment variable T by a randomly generated variable (e.g., Gaussian).

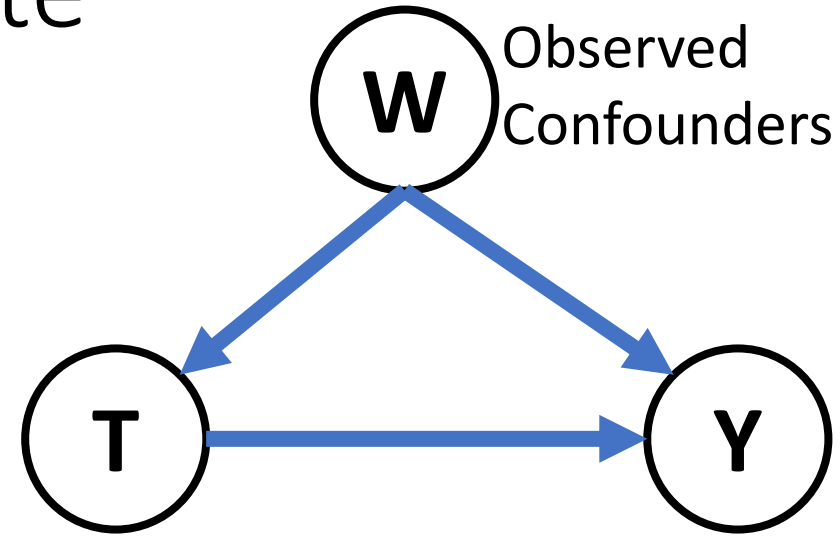
- Rerun the causal inference analysis.
- If the estimate is significantly away from zero, then analysis is incorrect.



Example 2: Add Unobserved Confounder to check sensitivity of an estimate

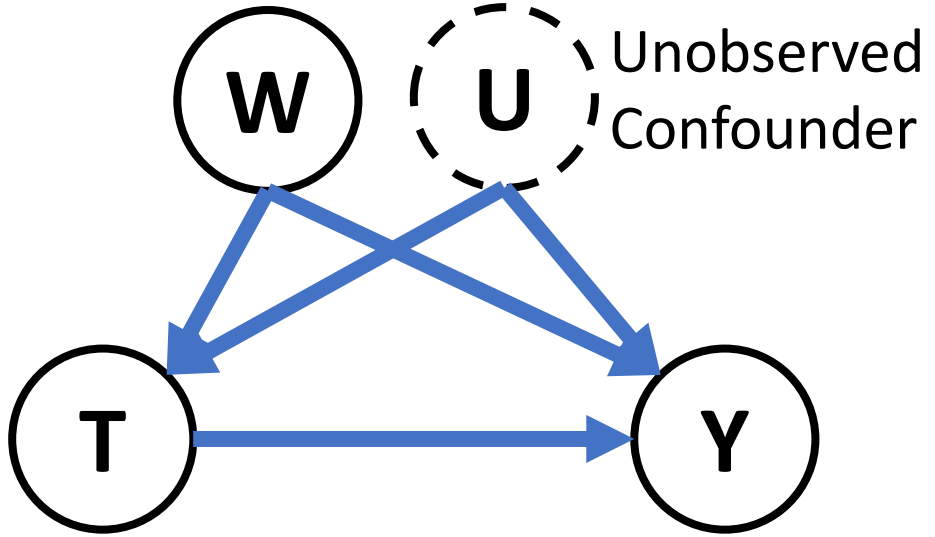
Q: *What if there was an unobserved confounder that was not included in the causal model?*

Check how sensitive the obtained estimate is after introducing a new confounder.



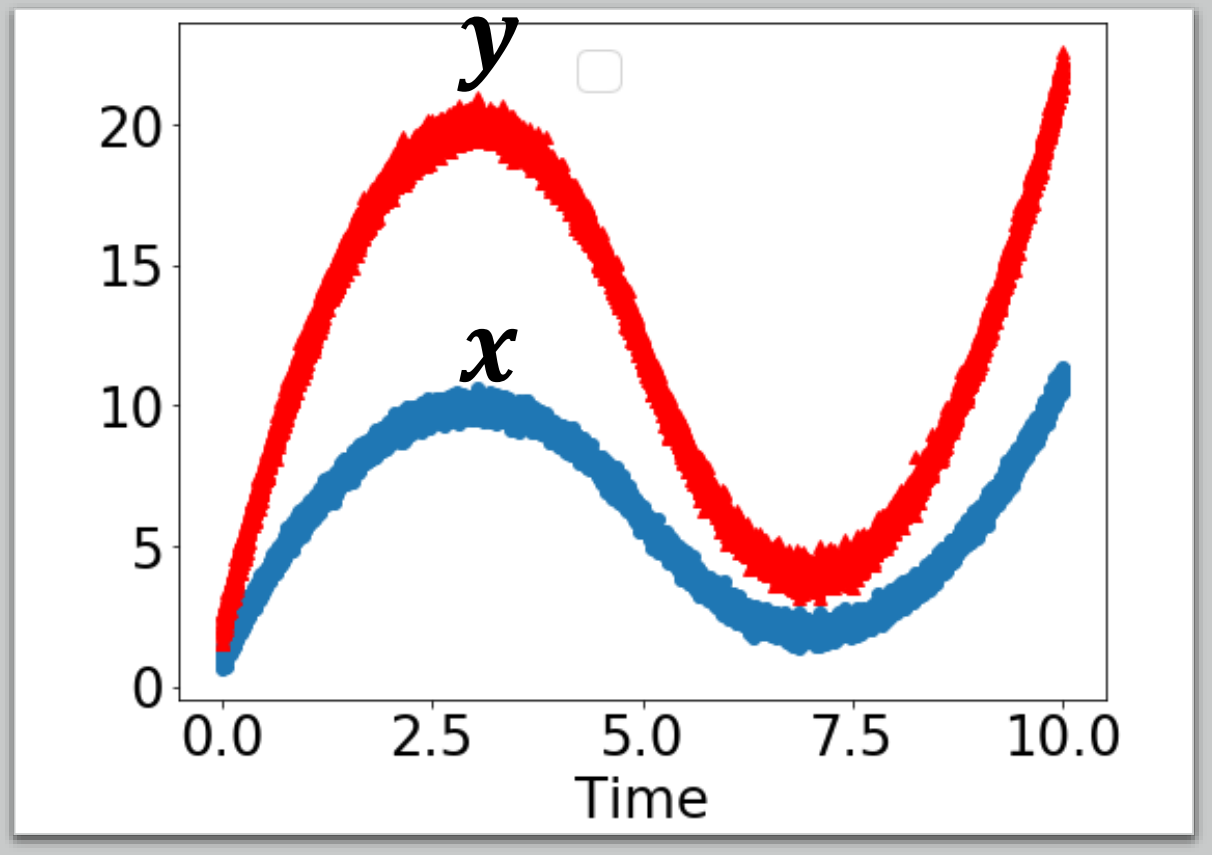
Unobserved Confounder Refuter:

- Simulate a confounder based on a given correlation ρ with both treatment and outcome.
  - Maximum Correlation ρ is based on the maximum correlation of any observed confounder.
- Re-run the analysis and check if the sign/direction of estimate flips.



Walk-through of the 4 steps using the DoWhy Python library

A mystery problem of two correlated variables: **Does x cause y?**



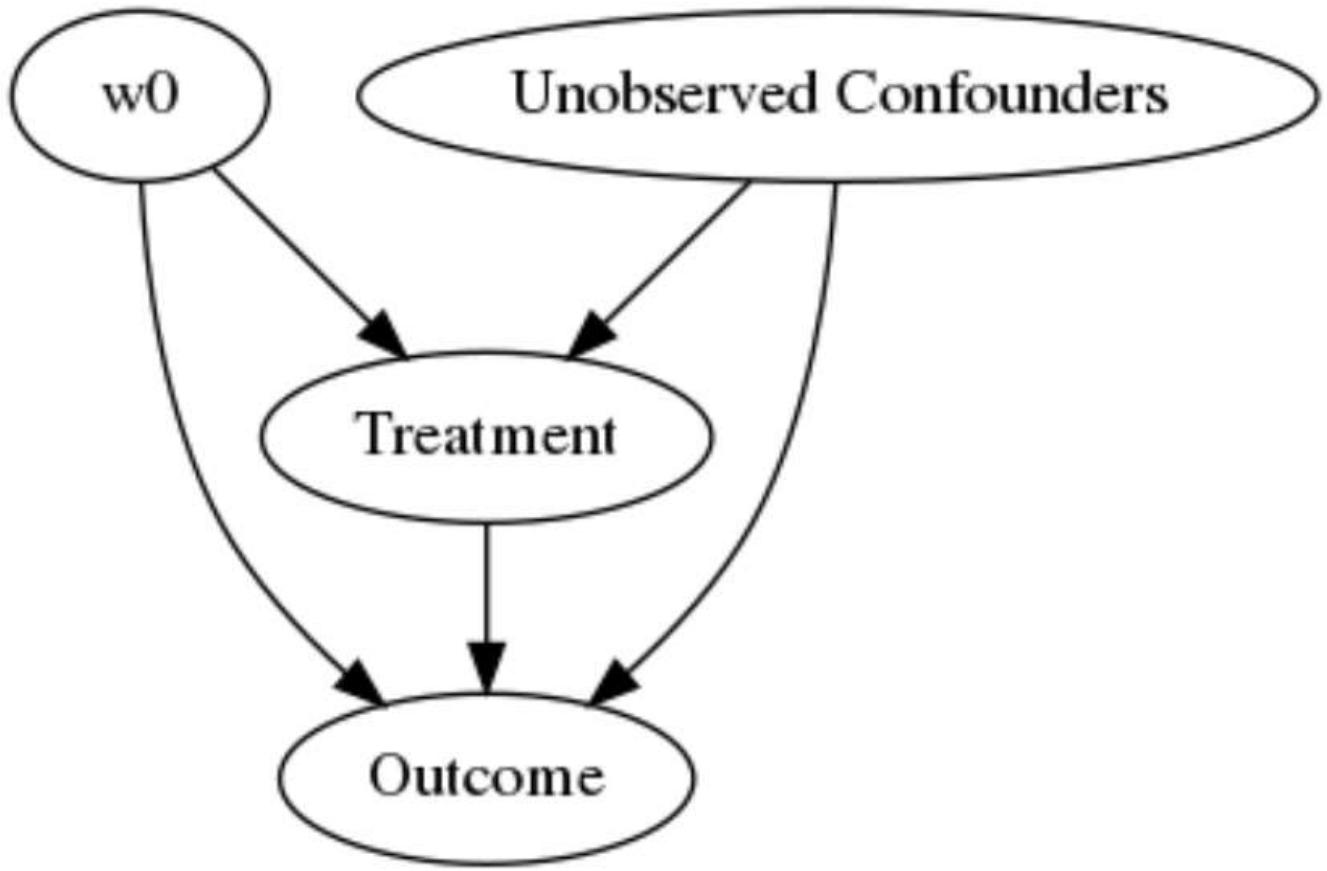
x	y	w
8.329680	16.546904	2.546634
2.083811	4.096492	-3.995819
6.138014	12.041800	0.041479
8.874336	17.833621	2.988497
5.282355	10.481077	-0.860686

STEP 1: Model the problem as a causal graph

Initializing the causal model.

In [5]:
```
model= CausalModel(
        data=df,
        treatment=data_dict["treatment_name"],
        outcome=data_dict["outcome_name"],
        common_causes=data_dict["common_causes_names"],
        instruments=data_dict["instrument_names"])
model.view_model(layout="dot")
```

In [6]:
```
from IPython.display import Image, display
display(Image(filename="causal_model.png"))
```



STEP 2: Identify causal effect using properties of the formal causal graph

Identify the causal effect using properties of the causal graph.

In [7]:
```
identified_estimand = model.identify_effect()
print(identified_estimand)
```

STEP 3: Estimate the causal effect

Once we have identified the estimand, we can use any statistical method to estimate the causal effect.

Let's use Linear Regression for simplicity.

In [8]:
```
estimate = model.estimate_effect(identified_estimand,
        method_name="backdoor.linear_regression")
print("Causal Estimate is " + str(estimate.value))
```

Step 4: Refuting the estimate

We can also refute the estimate to check its robustness to assumptions (*aka* sensitivity analysis, but on steroids).

Replacing treatment with a random (placebo) variable

In [11]:
```
res_placebo=model.refute_estimate(identified_estimand, estimate,
        method_name="placebo_treatment_refuter", placebo_type="permute")
print(res_placebo)
```

```
INFO:dowhy.causal_estimator:INFO: Using Linear Regression Estimator
INFO:dowhy.causal_estimator:b: Outcome~placebo+w0

Refute: Use a Placebo Treatment
Estimated effect:(1.0154712956668286,)
New effect:(-0.00121214336331476 6,)
```

You can try out this example on Github:
https://github.com/microsoft/dowhy/blob/master/docs/source/example_notebooks/dowhy_confounder_example.ipynb

What else is in DoWhy?

- A unified, extensible API for causal inference that allows external implementations for the 4 steps
  - Can use estimation methods from external libraries such as **EconML** and **CausalML**.

```
dml_estimate = model.estimate_effect(identified_estimand,
                                     method_name="backdoor.econml.dml.DMLCateEstimator",
                                     target_units = lambda df: df["X0"]>1,
                                     confidence_intervals=True,
```

- A convenient CausalDataFrame (contributed by Adam Kelleher)
  - Pandas DataFrame extension with inbuilt functions for calculating causal effect.

Summary: DoWhy, a library that focuses on **causal assumptions and their validation**

Goal: A unified API for causal inference problems, just like PyTorch or Tensorflow for predictive ML.

Growing open-source community: > 30 contributors

- Roadmap: More powerful refutation tests, counterfactual prediction.
- Please contribute! Would love to hear your ideas on Github.

Resources

- DoWhy Library: https://github.com/microsoft/dowhy
- Arxiv paper on the four steps: https://arxiv.org/abs/2011.04216
- Upcoming book on causality and ML: http://causalinference.gitlab.io/

thank you– Amit Sharma (@amt_shrma)