Causal Machine Learning in Practice

Intro to DoWhy and DiCE libraries

Amit Sharma

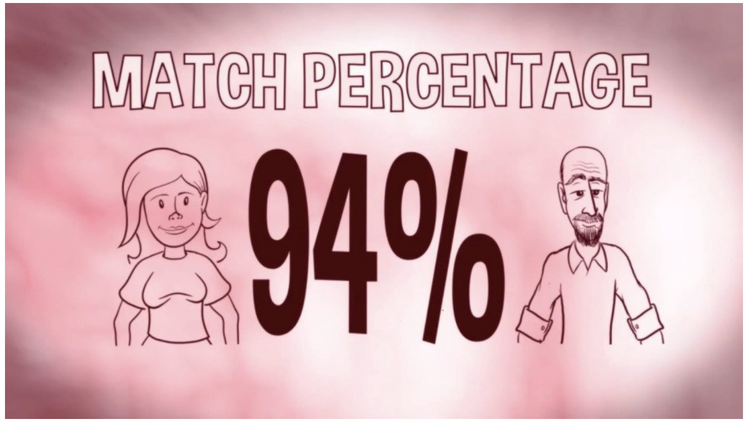
Principal Researcher

Microsoft Research India

Twitter: @amt_shrma

www.amitsharma.in

When we think of machine learning, we often think of **predictions: What does the data say?**



But there’s an important class of problems about **decisions: what action should I take?**

Which customers should we provide discounts to improve sales?

Which treatment will have the best improvement for a patient?

What is the best way to share an important public safety message?

Would this government regulation lead to a decrease in air pollution?

Sometimes, these problems overlap…

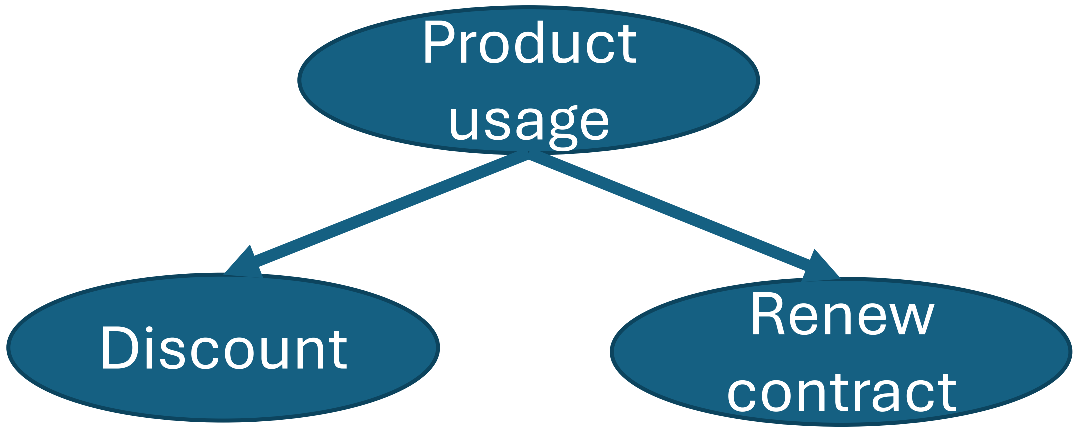
- Accurate prediction also means accurate decision-making.

- **Prediction task:** Does the X-ray image indicate a tumor?
- **Decision task:** Should we give tumor treatment or not?

But sometimes, they do not

- **Prediction task:** Predict the customers most likely to churn out.
- **Decision task:** Who to provide discounts to?
  - Discounts may not work on people likely to churn out (low activity)
  - May be unnecessary for people with high activity.
  - **Only need to find the people in the middle,** who are undecided.

Reason: Correlation versus causation



- Today’s product usage can predict tomorrow’s probability of churn (not renewing contract).
- But does not tell us anything about effect of discount.
  - Effect could even be zero!

And often, decision-making requires solving a new kind of problem: **effect estimation**

- **Effect estimation:** What is the effect of an action on the outcome?

What is the best way to share an important public safety message?

Q: What is the effect of sharing medium on response rate for the safety message?

Would this government regulation lead to a decrease in air pollution?

Q: What is the effect of the regulation on air pollution?

In effect estimation, the most important task is **how to avoid being fooled by correlations**

What is the best way to share an important public safety message?

Observed data: The response rate of text messages is the highest.

Selection bias: Dataset contains mostly young people.

Would this government regulation lead to a decrease in air pollution?

Observed data: In other states, pollution decreased after the regulation.

Confounding bias: Other states differ on the kind of industries they have.

So, how to solve these problems in a systematic way?

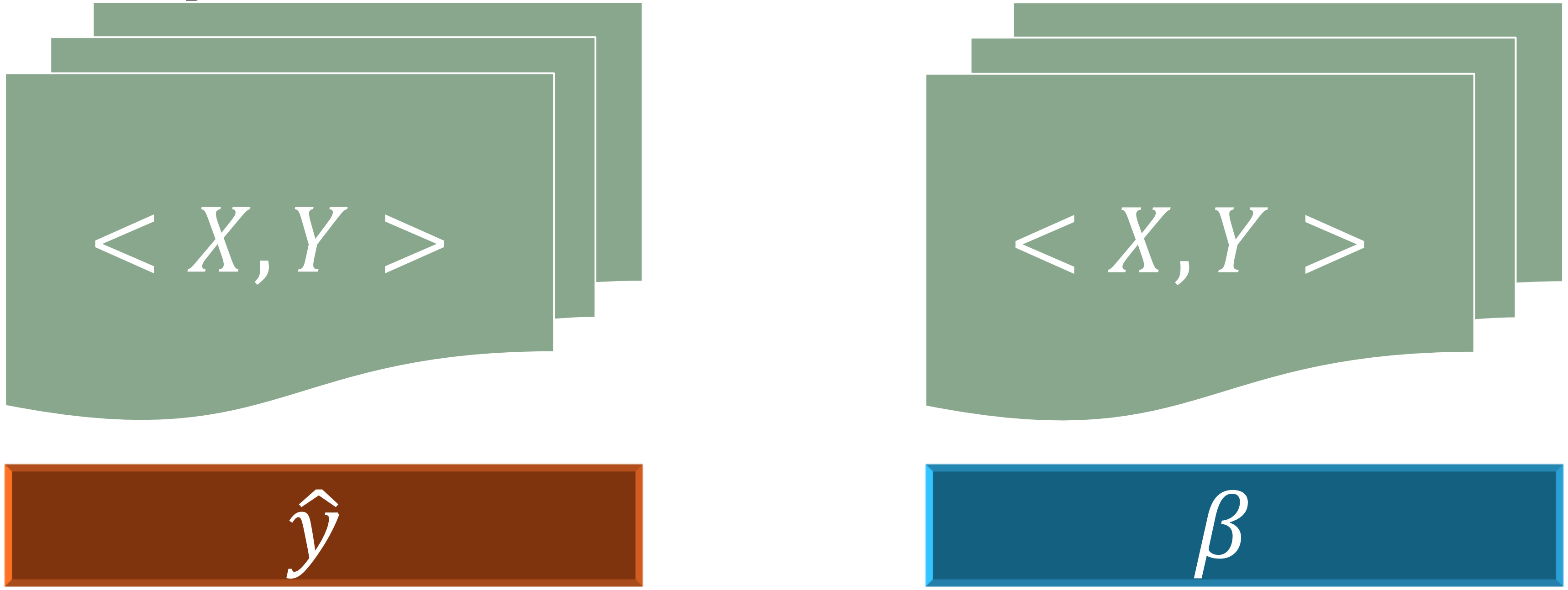
Incorporate techniques for **learning causality** in ML models.

Causal ML is about inferring the **best actions** (and the effects of actions in general)

True:
y=f(x,u)+ϵ

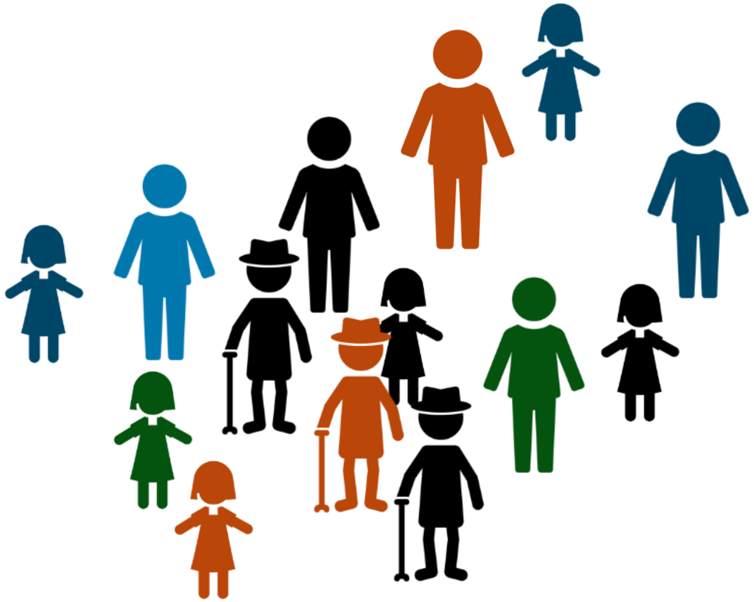
Prediction ML:
y=h(x)+ϵ

Causal ML:
$\partial f(x,u)/\partial x$



Hofman, **Sharma**, and Watts (2017). Prediction and Explanation in Social Systems. *Science*, 355.6324

Three key applications of causal ML: **Better decision-making** (what to do next?)

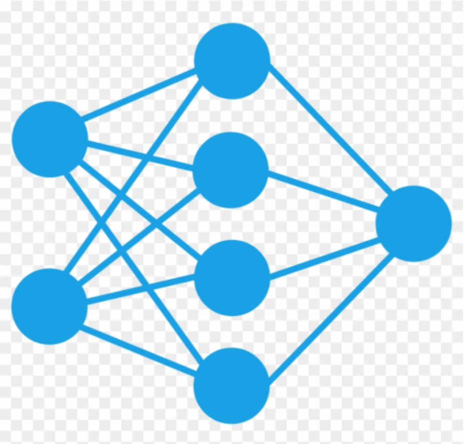


People who do not cycle have high cholesterol

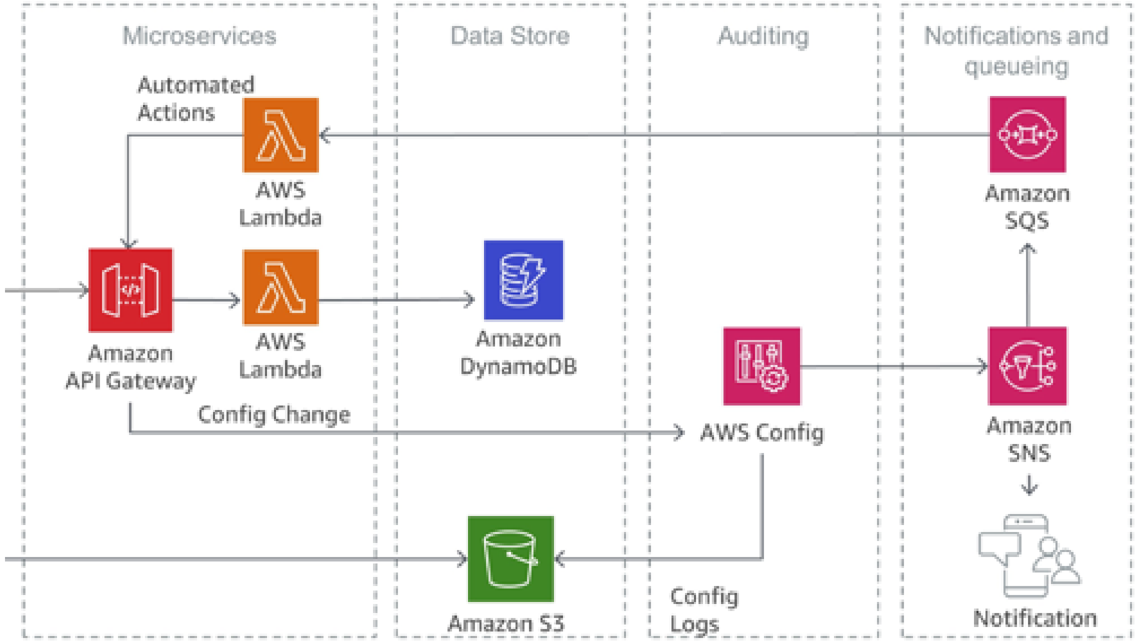
People who cycle regularly have low cholesterol

Decision: To improve cholesterol levels of the population, should the city government invest in programs for encouraging cycling (e.g., giving free bikes)?

Three key applications of causal ML: **Root cause attribution** (why did this happen?)



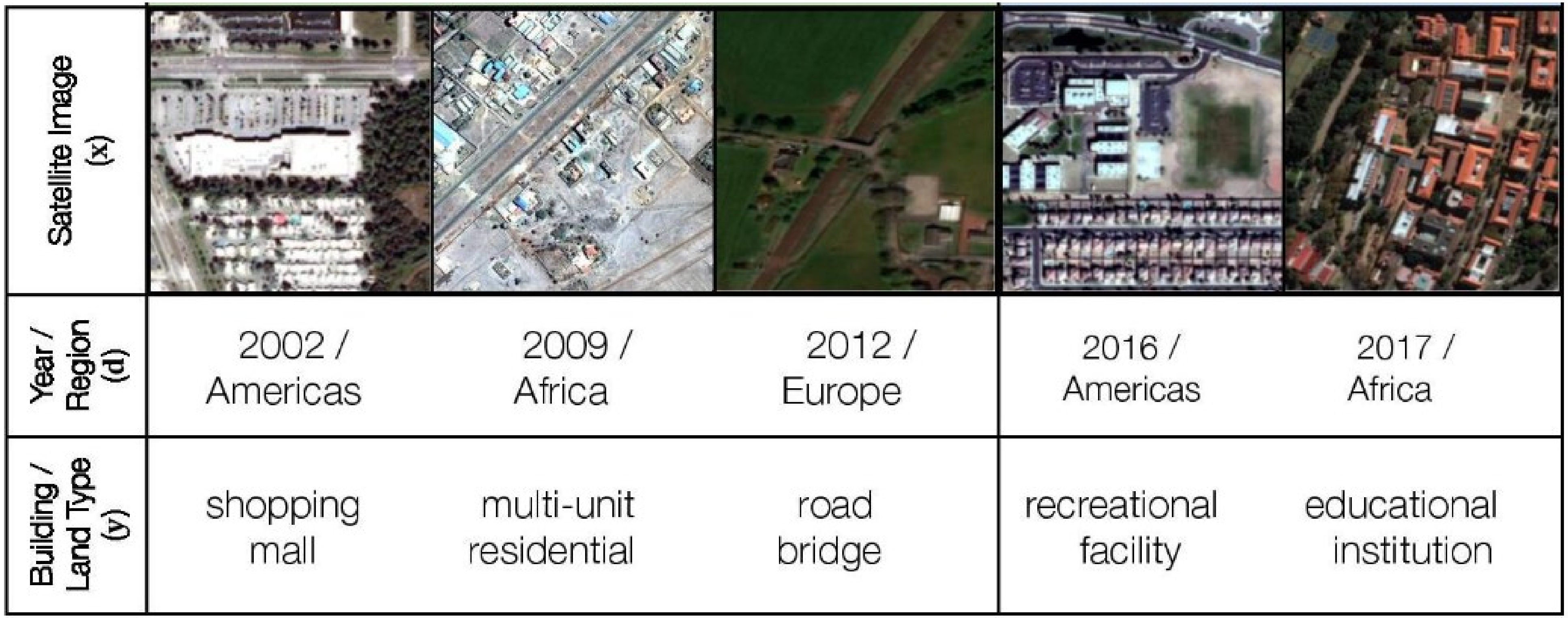
Predicted Class
Class: 1
Class: 0
Class: 0
Class: 1



Attribution: Why did the classifier predict Class:1 for the first image?

Attribution: For a given the microservice system, why did the latency increase?

Three key applications of causal ML: **Out-of-distribution generalization**



Koh et al., WILDS, ICML 2021

To summarize,

Causal ML:

Machine learning + causality

A necessary ingredient for general-purpose AI

- Effect inference (DoWhy)
- ML model attribution (DiCE)

Part I:

DoWhy: Estimating causal effects

https://github.com/py-why/dowhy

The four key 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.

To implement these 4 steps, we built DoWhy, an open-source library for causal inference

DoWhy makes assumptions front-and-center of any analysis

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

Most popular causal library on GitHub (>2M downloads, 7k stars)

  - Taught in third-party tutorials and courses: O'Reilly, PyData, Northeastern, …

- Open-source community: > 60 contributors
  - Including major contributions: Amazon, CMU, Columbia, etc.

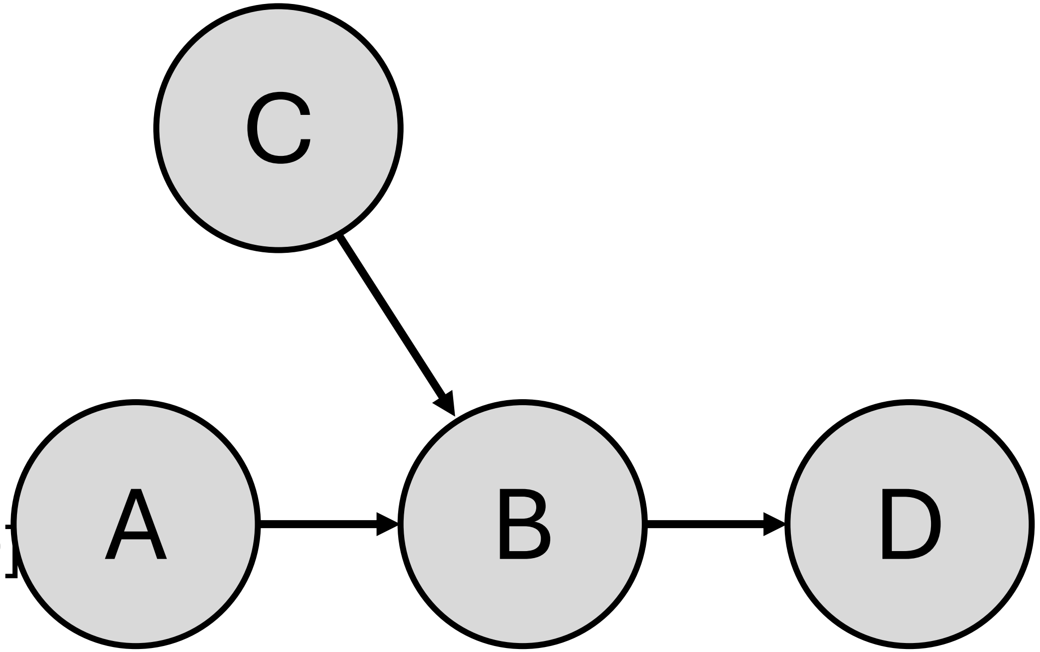
Goal: An end-to-end platform for doing causal inference

I. Model the assumptions using a causal graph

Convert domain knowledge to a formal model of causal assumptions

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

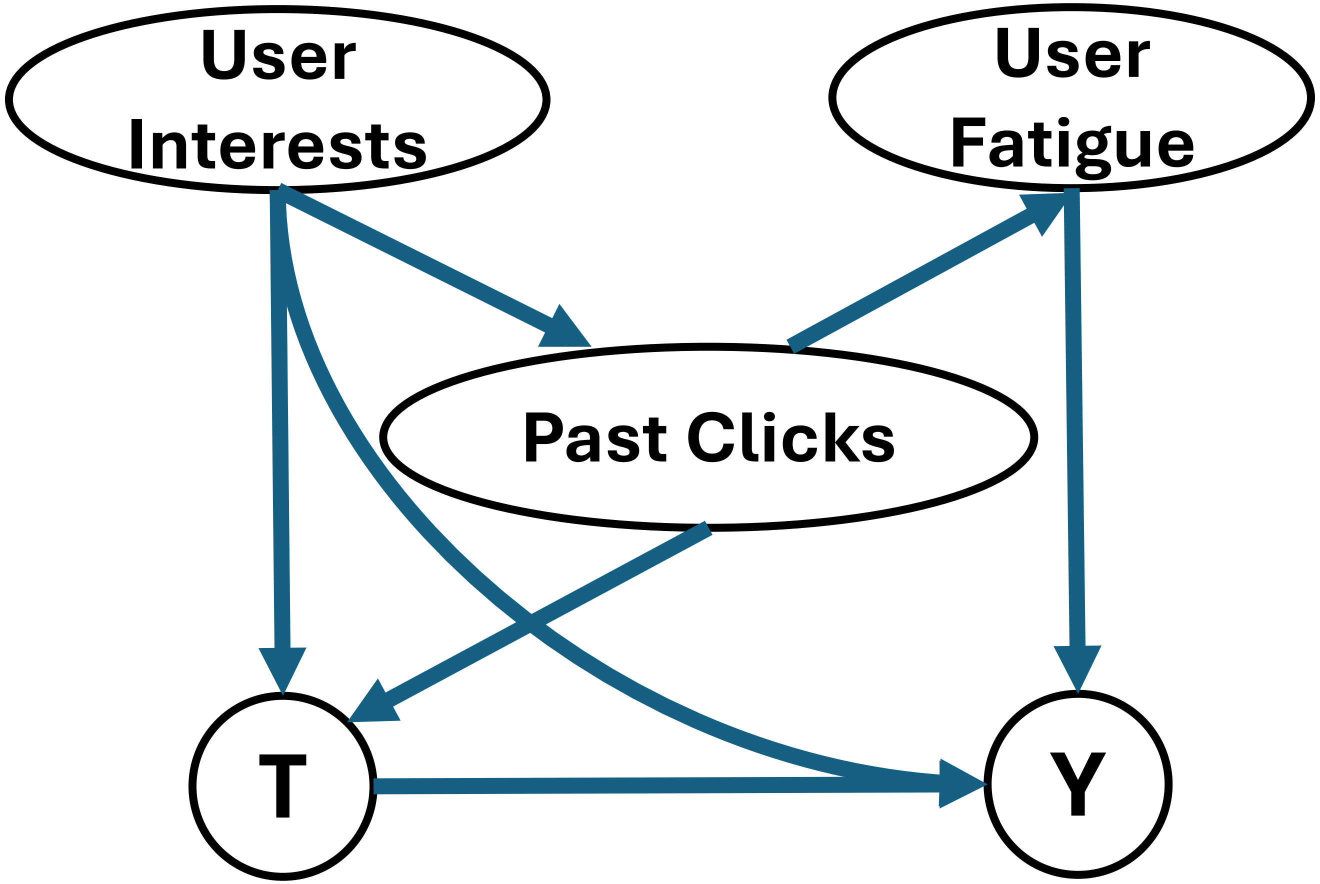
- Causal graph implies conditional statistical independences
  - E.g., $A \perp\!\!\!\perp C,\ D \perp\!\!\!\perp \mathrm{A} \mid \mathrm{B}, \ldots$
  - Identified by *d-separation* rules [Pearl 2009]
- These assumptions significantly impact the causal estimate we'll obtain.



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
  - Need not model everything

Example Graph



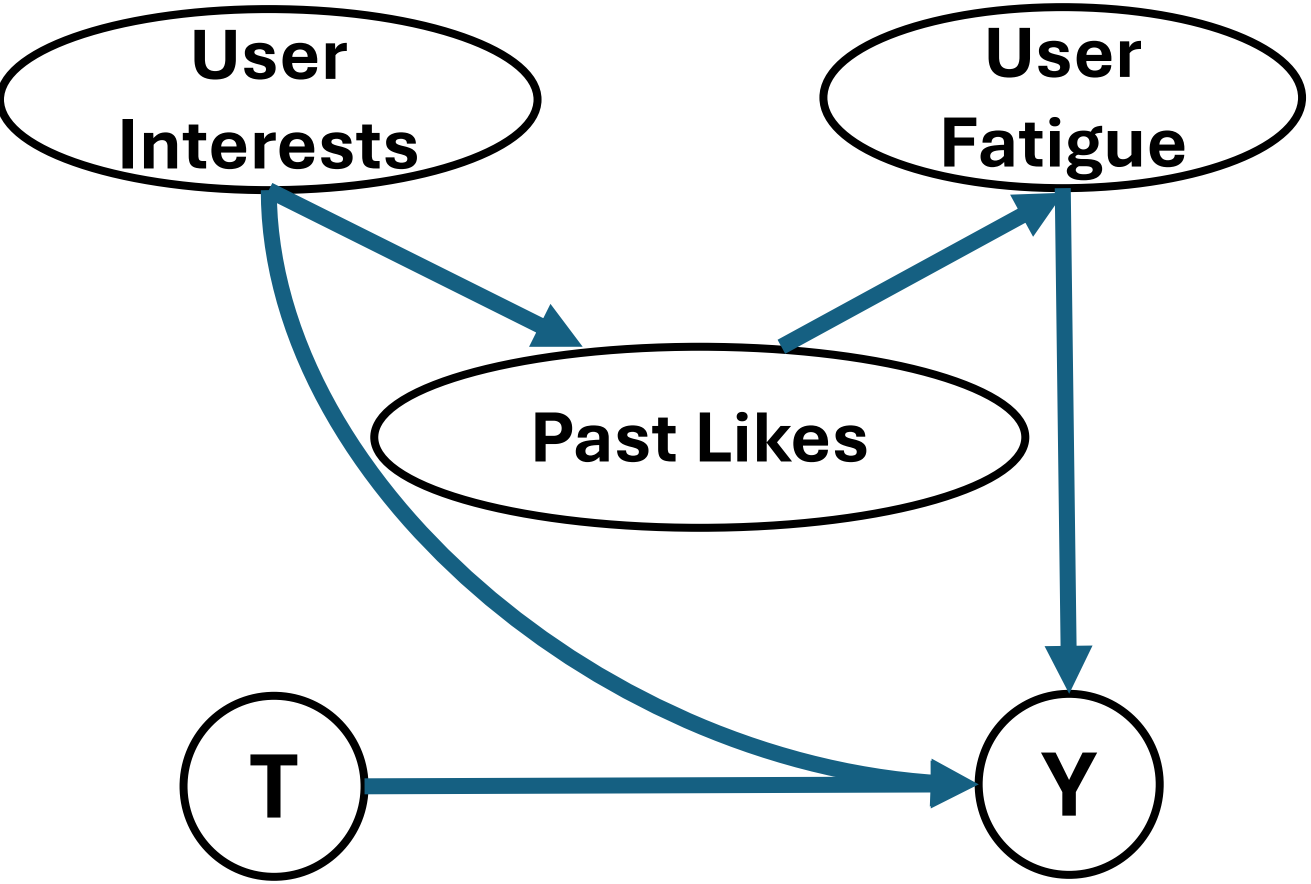
Assumption 1: User fatigue does not affect user interests

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



Interventional graph:

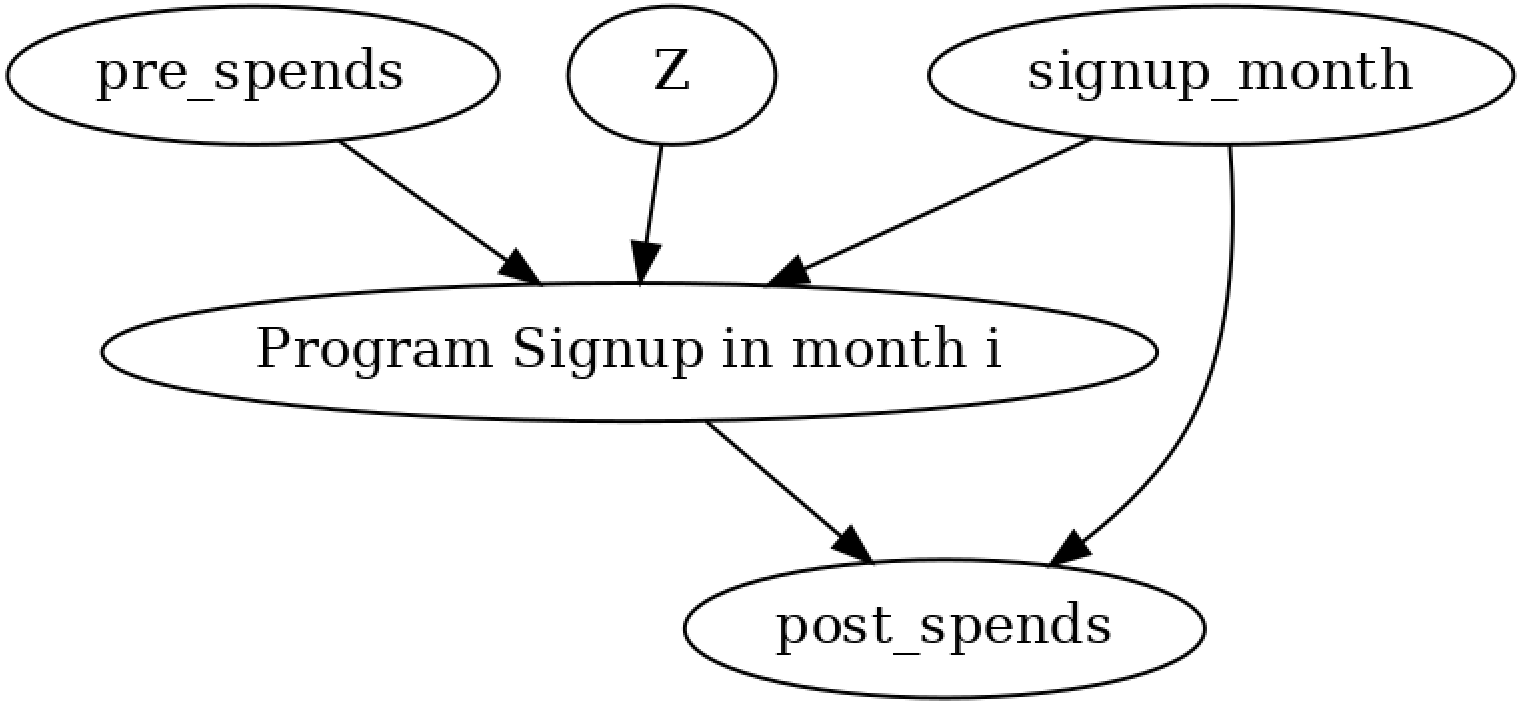
All edges to Treatment T removed, *keeping everything else the same*.

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

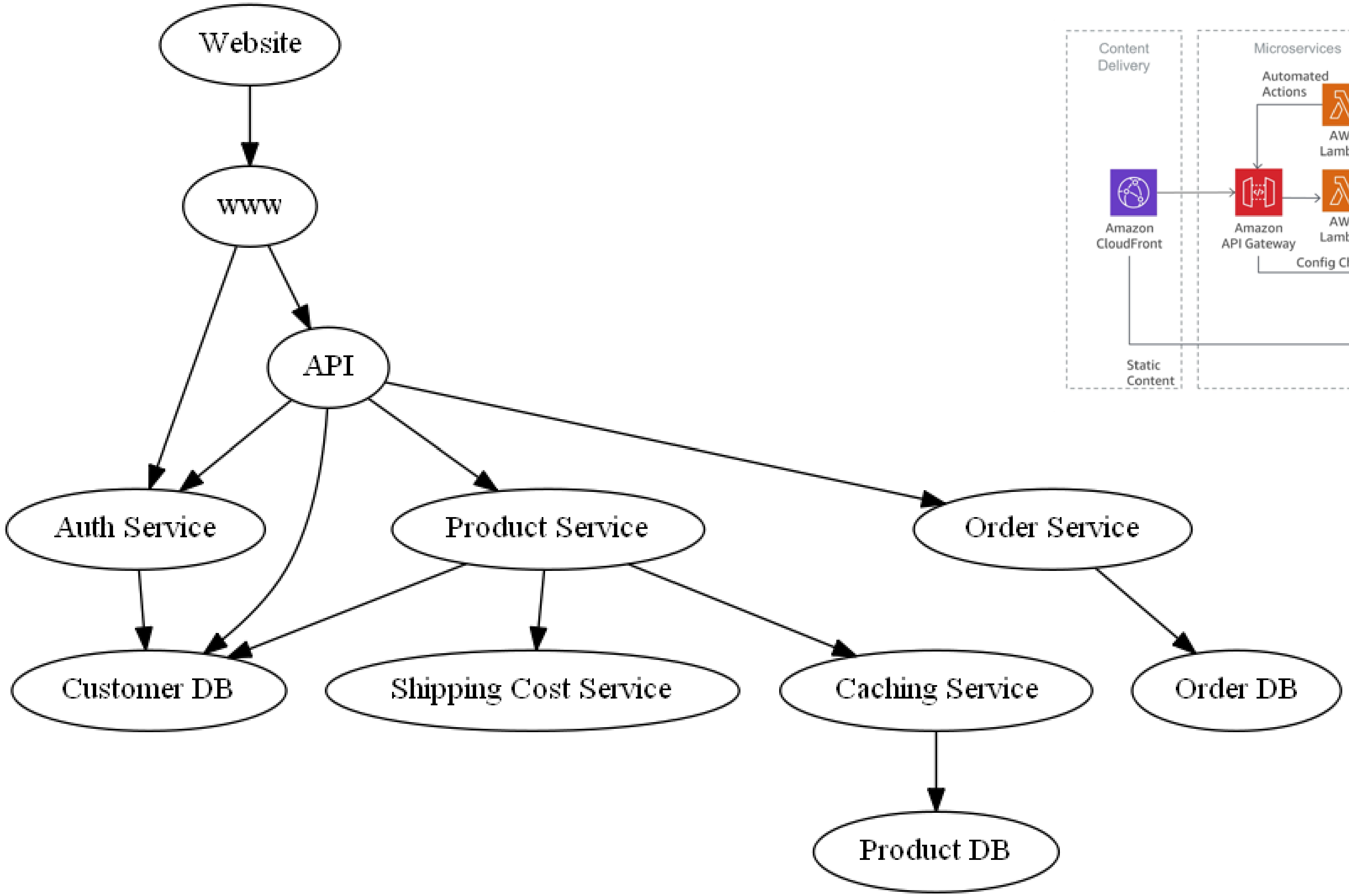
Causal effect: $P(Y|do(T))$

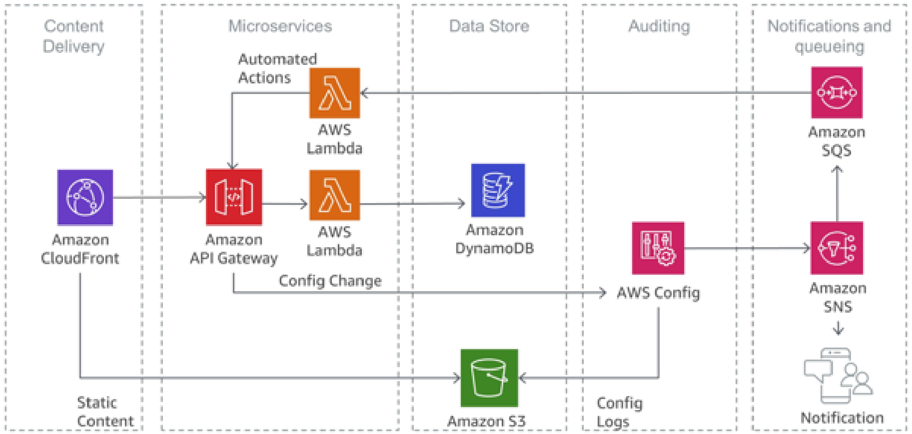
How to obtain a causal graph?
Use **domain knowledge** (Example 1)

- Estimating the effect of customer rewards program



How to obtain a causal graph?
Use **system knowledge** (Example 2)





How to obtain a causal graph?
Use **LLMs' world knowledge** (Example 3)

USER

You are a helpful assistant to a neuropathic pain diagnosis expert. Which cause-and-effect relationship is more likely?

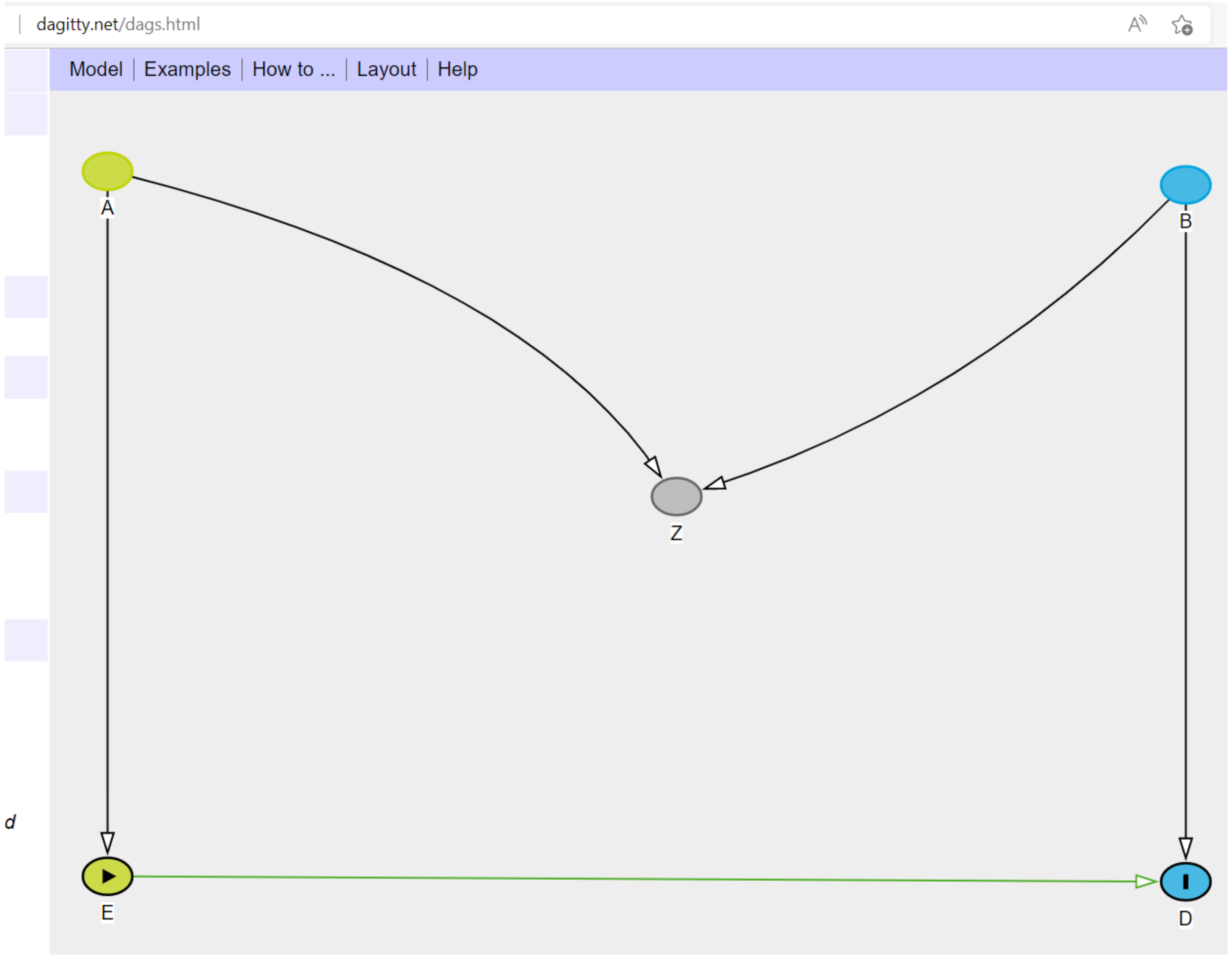
A. Left T6 Radiculopathy causes DLS T5-T6.
B. DLS T5-T6 causes Left T6 Radiculopathy.

Let's work this out in a step by step way to be sure that we have the right answer. Then provide your final answer within the tags <Answer>A/B</Answer>.

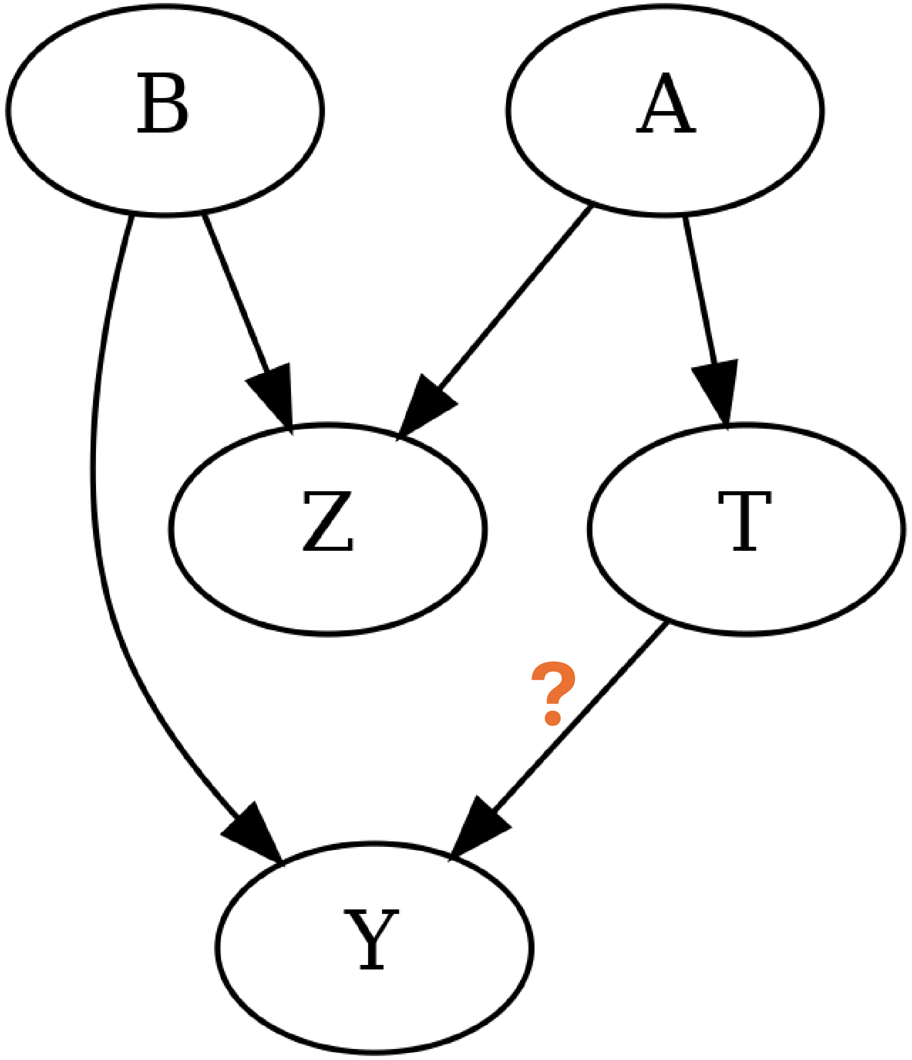
GPT-4

First, we need to understand the terms involved in the cause-and-effect relationship.
Left T6 Radiculopathy refers to a condition where there is damage or irritation to the nerve root that exits the spinal cord at the level of T6 on the left side of the body.
DLS T5-T6 refers to degenerative changes in the discs, ligaments, and/or bones at the level of T5-T6 in the spine.
Now, let's consider the two options:
A. Left T6 Radiculopathy causes DLS T5-T6. This would mean that the damage or irritation to the nerve root at T6 on the left side of the body somehow leads to degenerative changes in the discs, ligaments, and/or bones at the level of T5-T6. While it is possible that nerve damage could contribute to degenerative changes in the spine, it is more likely that the opposite is true.
B. DLS T5-T6 causes Left T6 Radiculopathy. This option suggests that the degenerative changes in the discs, ligaments, and/or bones at the level of T5-T6 are causing the Left T6 Radiculopathy. This is a more likely cause-and-effect relationship, as degenerative changes in the spine can lead to nerve root irritation or compression.
Therefore, the more likely cause-and-effect relationship is: <Answer>B</Answer>DLS T5-T6 causes Left T6 Radiculopathy.

Example: *LLM correctly describes medical terms and can identify the causal direction, with some error.*



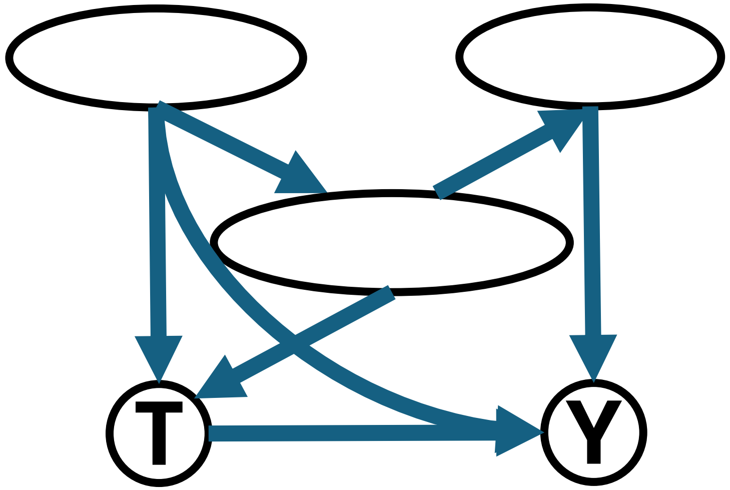
DAGitty.net



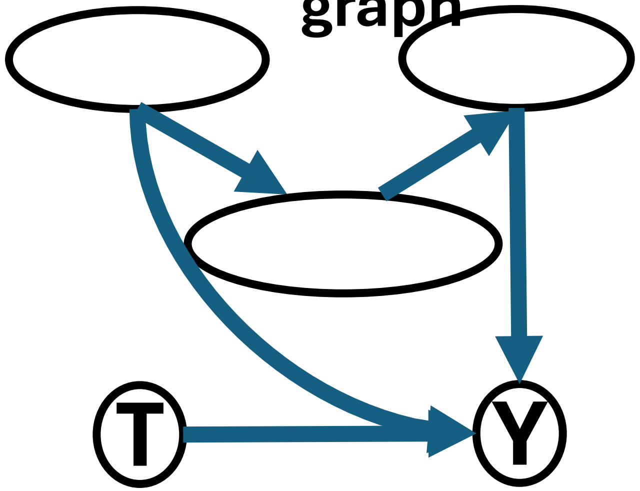
DoWhy

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

Observed data generated by this graph



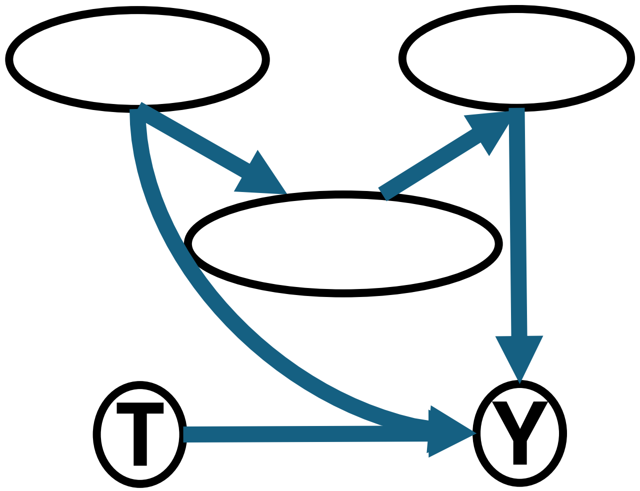
Want to answer questions about data that *will* be generated by intervention graph



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



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 or other criteria*

Intuitions:

- The union of all features is *not* necessarily a valid adjustment set
  - Depends on the graph structure and unobserved variables

So far, we have not used any dataset!

- Given a graph, identification of causal effect does not require access to data.
  - (sometimes requires parametric assumptions)
- Important to distinguish between identification and estimation
  - DoWhy has two separate API calls
    - Identify_effect()
    - Estimate_effect()
- Provides clean separation of assumptions
  - Some assumptions during identification (e.g., no unobserved confounder)
  - Others during estimation (e.g., linear model)

Many kinds of identification methods

Graphical constraint-based methods

- Adjustment Sets
    - Backdoor, "towards necessity"
- ID algorithm
- Front-door criterion
- Mediation formula

Identification under additional non-graphical constraints

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

All 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.

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

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)

Example: Estimating the effect of a customer loyalty rewards program

What is the impact of offering the customer loyalty program on total sales?

If the current members ***had not signed up*** for the program, how much less would they have spent?

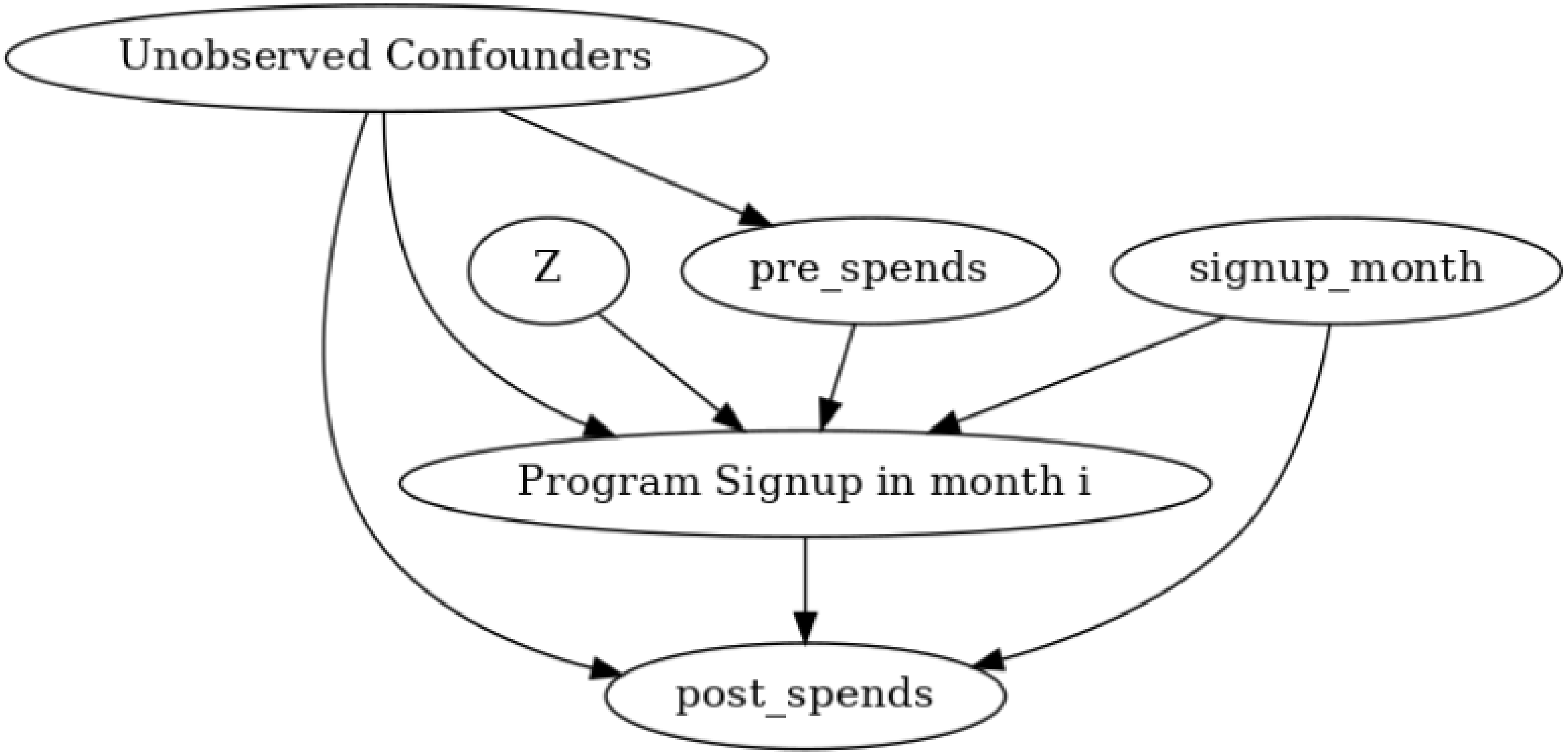
ATT: *Average treatment effect on the treated* (customers who signed up for the program)

	user_id	signup_month	month	spend	treatment
0	0	6	1	507	True
1	0	6	2	506	True
2	0	6	3	490	True
3	0	6	4	464	True
4	0	6	5	475	True
...	...	...	...	...	...
119995	9999	0	8	396	False
119996	9999	0	9	387	False
119997	9999	0	10	367	False
119998	9999	0	11	436	False

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

Step 1: Modeling. Create causal graph to encode assumptions.

```
model = dowhy.CausalModel(data=df_i_signupmonth,
                          graph=causal_graph.replace("\n", " "),
                          treatment="treatment",
                          outcome="post_spends")
```



Step 2: Identification. Formulate what to estimate

```
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
print(identified_estimand)
```

Step 3: Estimation. Compute the estimate

```
estimate = model.estimate_effect(identified_estimand,
                                  method_name="backdoor.propensity_score_matching",
                                 target_units="att")
print(estimate)
```

Wait, how do we know the estimate is *correct*?

Need causal validation tests.

IV. Refutation tests: 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?

Ways to validate a causal estimate

“INTEGRATION” TESTS

- Test the entire analysis pipeline.
- Run a randomized trial
- Use “Negative” controls [Lipsitch et al. 2010]
- Use sensitivity analysis

“UNIT” TESTS

- Test a specific step of the pipeline.
- Conditional Independence test
- Bootstrap / Data subset test

Called “refutations” because they can invalidate a bad analysis, but cannot prove that an analysis is correct.

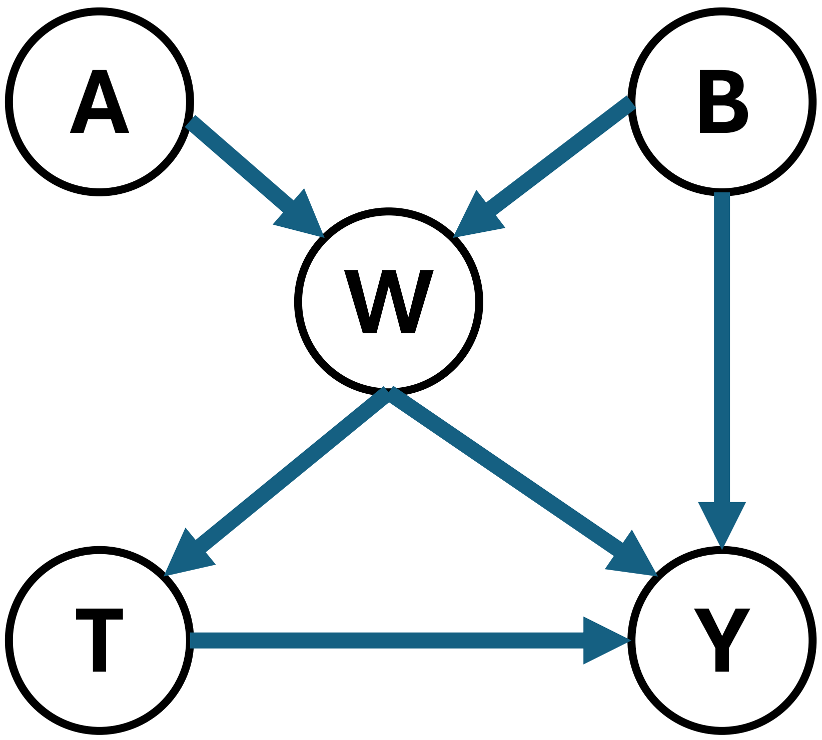
Unit refuter 1: Conditional Independence

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:

$A \perp\!\!\!\perp B$ $\quad$ $A \perp\!\!\!\perp T | W$ $\quad$ $B \perp\!\!\!\perp T | W$

Integration test: Negative control

- A concept from biology and physical sciences
  - Suppose you obtain a positive effect.
  - Re-run an experiment without a necessary condition (e.g., main reacting agent)
  - If effect does not go to zero, then the experimental setup is incorrect.

For causal inference

- Suppose you obtain a significant effect.
- Construct a new dataset where true causal effect is known (e.g., zero).
- Re-run the analysis on the dataset and check if it matches the true value.

Integration test: Placebo Treatment (“A/A”)

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

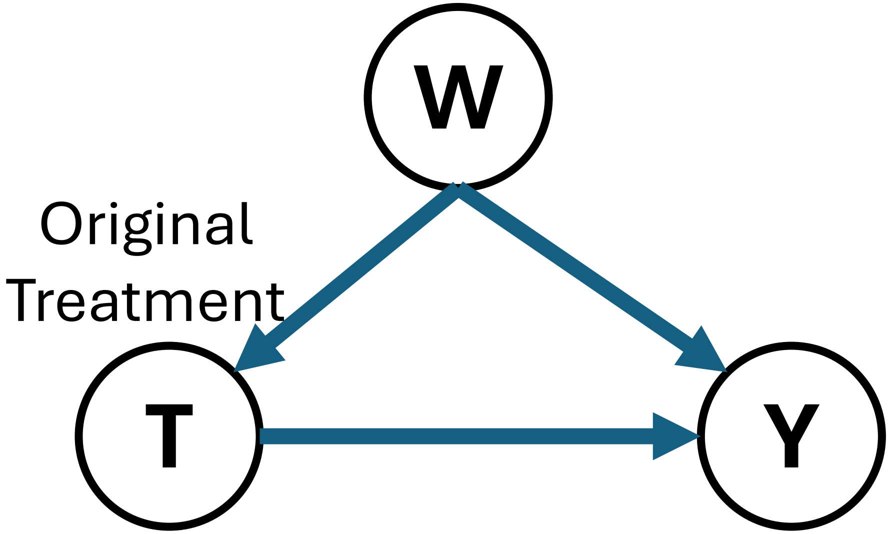
Then a correct causal inference method should return an estimate of zero.

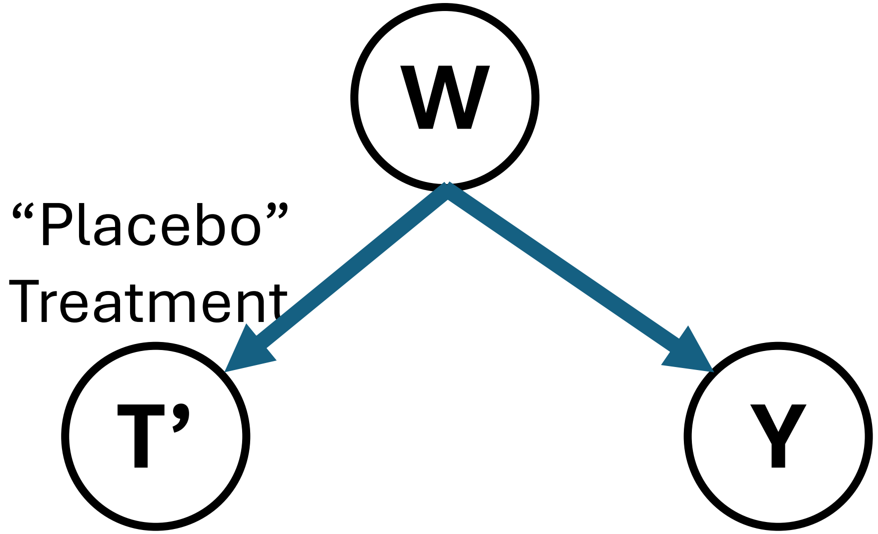
Placebo Treatment Refuter:

Pick a variable that is known not to cause the outcome. ***OR***

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.

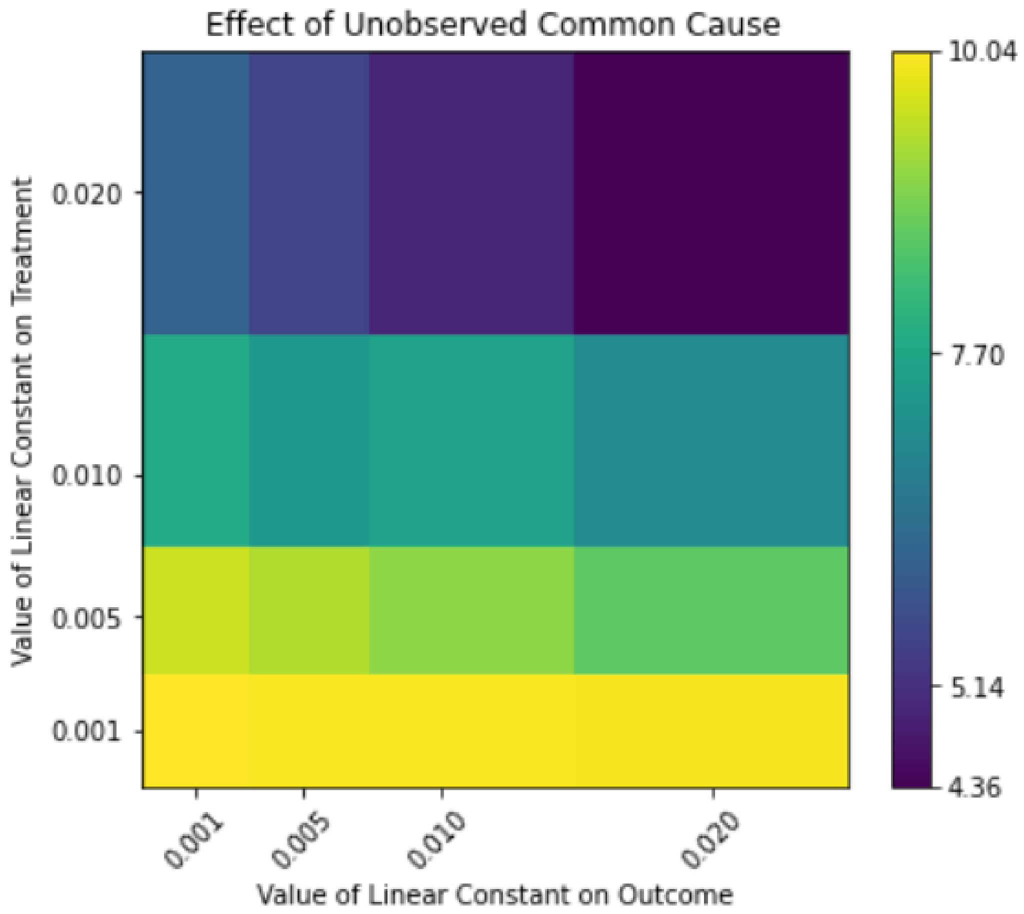




Step 4: Refutation. Validate the assumptions

```
refutation = model.refute_estimate(identified_estimand, estimate, method_name="placebo_treatment_refuter",
                    placebo_type="permute", num_simulations=20)
print(refutation)
```

```
Refute: Use a Placebo Treatment
Estimated effect:100.03963044006804
New effect:0.6054947726720156
p value:0.24154316295878647
```



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.*

Can we enable new tasks beyond effect inference using the same 4 steps?

Py-Why GitHub organization: DoWhy, EconML, causal-learn, pywhy-llm

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

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

Growing open-source community: > 40 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)

Part II:

DiCE: Explaining machine learning models using counterfactuals

https://github.com/interpretml/dice

Assessing human decision-making

Sandra Bauer Meryem Öztürk

Are employers in Germany biased against women wearing a hijab?
[Weichselbaumer 2019]

Sandra Bauer Meryem Öztürk

Meryem Öztürk Meryem Öztürk

Assessing human decision-making

Sandra Bauer Meryem Öztürk

Counterfactual reasoning has been used the social sciences to assess different aspects of human decision-making [Bertrand and Mullainathan 2003, Weichselbaumer 2019]

Sandra Bauer Meryem Öztürk

Meryem Öztürk Meryem Öztürk

Why does counterfactual reasoning work?

Because only the specific input is varied, provides the **causal effect** of the input, specific to the current context.

Also known as individual causal effect.

What is a counterfactual?

Given a system output y,
a counterfactual $y_{X_i=x'}$ is the output of the system had some input X_i changed but everything else unaffected by X_i remained the same.[Pearl 2009]

REAL WORLD

$(X_i = x)$

COUNTERFACTUAL WORLD

$(X_i = x')$

Counterfactual: $P\left(Y_{X_i=x'} \middle| \boldsymbol{X} = \boldsymbol{x}, Y = y\right)$

→Since a ML model f is a deterministic model, counterfactual simplifies to $f\left(\boldsymbol{X}_{X_i=x'}\right)$

The many uses of a model counterfactual

Individual Effect of Input Feature $\mathrm{X_i}$

$=E\left(Y_{X_i=x'} \middle| \boldsymbol{X}=\boldsymbol{x}, Y=y\right) - E(Y|\boldsymbol{X}=\boldsymbol{x})$

$f\left(\boldsymbol{X}_{X_i=x'}\right) - f(\boldsymbol{X})$ can provide:

1. Explanation of how important X_i feature is.
2. Bias in the model if X_i is a sensitive feature.
3. More generally, provides a natural way to debug ML models (*ala fuzz testing*).

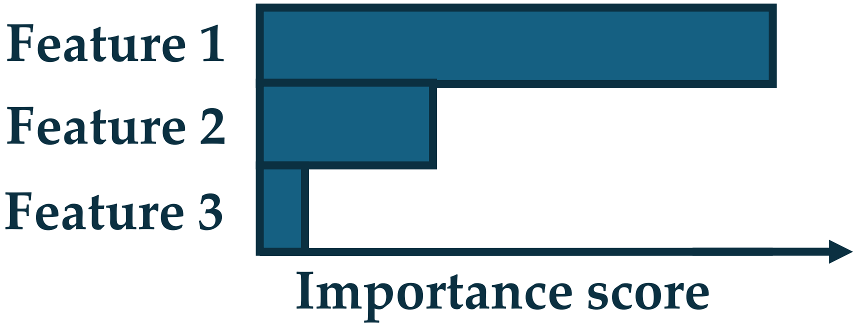
Why use counterfactuals when there are many established methods of ML model explanation?

Explaining machine learning predictions

Techniques to explain machine predictions

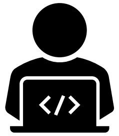
LIME (Ribeiro et al., 2016); Local Rule-based (Guidotti et al., 2018); SHAP (Lundberg et al., 2017); Intelligible Models (Lou et al., 2012); …..

Feature importance-based methods are widely used in many practical applications



In many cases, feature importance is not enough

Bank → Loan distribution model → Loan granted / Loan denied

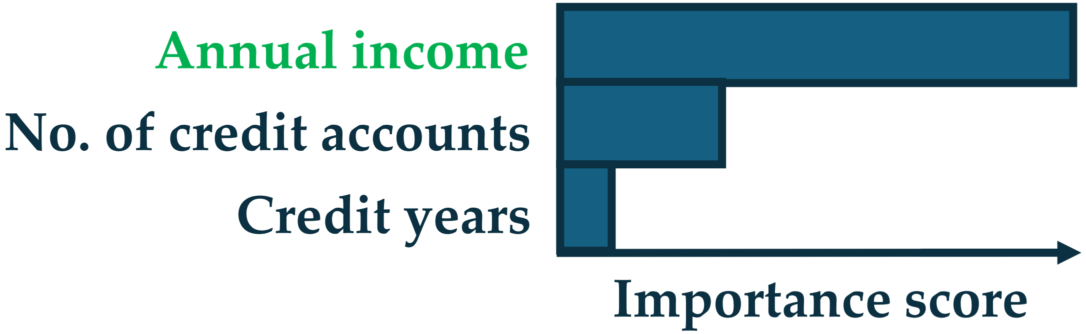


Suppose model predicts that the person should not get the loan.

Decision-maker: Why should this person not get the loan?

Person: What should I do to get the loan in the future?

Feature importance-based explanations

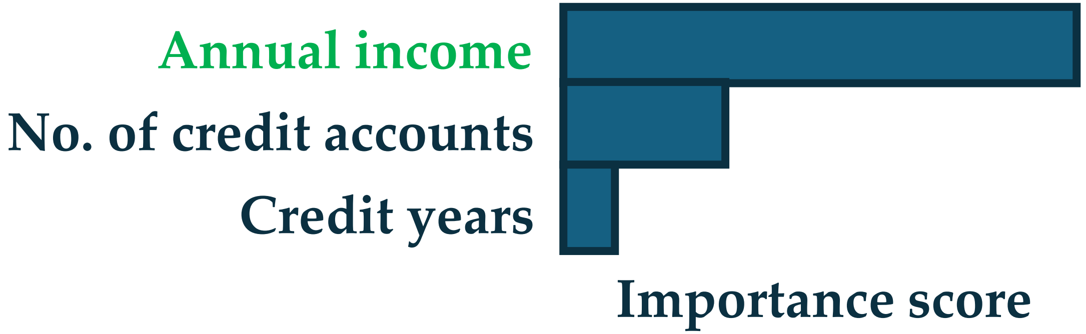


Counterfactual explanations (CF)

("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

Feature importance-based explanations



Interpretable,
but not high-fidelity

Counterfactual explanations (CF)

("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

Interpretable,
and high-fidelity

Catch: How to generate the right examples that are useful to end-user?

Desirable properties for counterfactuals

Actionability [Only for decision-subject] :
Users should be able to make the changes indicated by counterfactuals

Feasibility **+** **Diversity**

- ✓ Proximity
- ✓ User constraints
- ✓ Sparsity
- ✓ Causal constraints

Wachter et al (2017)

$$C = \arg\min_{c} yloss(f(c), y) + |x - c|$$

Russell (2017)
Mixed integer programming
Works only for linear ML models

General optimization framework

Diverse counterfactual explanations

Loss to get **desirable outcome**

Loss to ensure **proximity** to original input

Loss to provide **diverse** explanations

$$\boldsymbol{C(x)} = \arg\min_{c_1,\ldots,\, ck} \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\boldsymbol{dist}(c_i, x) - \lambda_2 \boldsymbol{dpp\_diversity}(c_1, .., c_k)$$

k – no. of counterfactuals

λ_1 and λ_2 – loss-balancing hyperparameters

$$\boldsymbol{dpp\_diversity} = \boldsymbol{det(K)},$$

$$\boldsymbol{K} = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{dist}(\mathbf{c}_i, \mathbf{c}_j)}$$

Diverse counterfactual explanations

Adult-Income:
Predicting income based on demographical and educational variables (UCI ML repository)

Adult	HrsWk	Education	Occupation	WorkClass	Race	AgeYrs	MaritalStat	Sex
Original input (outcome: <=50K)	45.0	HS-grad	Service	Private	White	22.0	Single	Female
Counterfactuals (outcome: >50K)	—	Masters	—	—	—	65.0	Married	Male
	—	Doctorate	—	Self-Employed	—	34.0	—	—
	33.0	—	White-Collar	—	—	47.0	Married	—
	57.0	Prof-school	—	—	—	—	Married	—

Counterfactual examples: A way to generate debugging edge-cases

Q. What is the minimum change in input features needed to change the model's prediction to higher income?

-- when changing only sensitive features (correct answer: Not possible)

-- when changing hours per week (correct answer: Only positive changes)

-- when changing all features (correct answer: Tiny changes should not matter)

ACM FaccT 2020. **Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations.** Mothilal et al.

How does DiCE compare to LIME and SHAP

- LIME and SHAP approximate sufficiency.
- DiCE (default) is a measure of necessity.

Example: $y = I(0.45x1 + 0.1x2 \geq 0.5)$

Low-ranked features from LIME/SHAP may be as powerful in changing the class.

Method	x_1	x_2
LIME	0.34	0.07
SHAP (median BG)	1.0	0.0
SHAP (train data BG)	0.69	0.28
DiCE_{FA}	0.975	0.967
$\text{WachterCF}_{\text{FA}}$	1.0	0.975

Table 1: Explaining model $y = I(0.45x_1 + 0.1x_2 \geq 0.5)$ at an input point $(x_1 = 1, x_2 = 1, y = 1)$. x_1 and x_2 are continuous features randomly sampled from an uniform distribution, $U(0, 1)$. The second and third column shows an explanation method's score for x_1 and x_2 respectively. For SHAP, the scores are shown for both median data and the entire training data as background (BG) sample in the second and third row respectively. Unlike attribution-based methods (LIME and SHAP), counterfactual-based methods (DiCE_{FA} and $\text{WachterCF}_{\text{FA}}$) give almost equal importance to x_2 feature even though its coefficient in the target model is much smaller than x_1's coefficient.

DiCE library: CFs in practice

Diverse Counterfactual Explanations

Practical considerations

$$\boldsymbol{C(x)} = \arg\min_{c_1,\ldots,c_k} \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\boldsymbol{dist}(c_i, x) - \lambda_2 \boldsymbol{dpp\_diversity}(c_1, \ldots, c_k)$$

- ❑ Incorporate additional feasibility properties
  a) Sparsity – post-hoc correction
  b) User constraints
- ❑ Choice of yloss – hinge loss
- ❑ Separate categorical and continuous distance functions
- ❑ Relative scale of mixed features

Python library
DiCE
(**D**iverse **C**ounterfactual **E**xplanations)
https://github.com/interpretml/DiCE

But what if the model is not differentiable?

Coming back to the optimization problem

Loss to get desirable outcome

Loss to ensure proximity to original input

Loss to provide diverse explanations

$$\boldsymbol{C(x)} = \arg\min_{c_1,\ldots,c_k} \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\boldsymbol{dist}(c_i, x) - \lambda_2 \boldsymbol{dpp\_diversity}(c_1, \ldots, c_k)$$

Given an input, generate new, proximal points such that they change the predicted class and are diverse.

Classic optimization problem

Can use any technique to sample points.

- **Random sampling**
  - Sample a lot of points, then filter according to criteria
  - Surprisingly not bad with low-dimensional features
  - Ensures coverage for all features
- **Genetic programming**
  - Construct a fitness function based on proximity and correctness
  - Keep exploring until reach a good value of fitness

```python
# Using sklearn backend
m = dice_ml.Model(model=model, backend="sklearn")
# Using method=random for generating CFs
exp = dice_ml.Dice(d, m, method="random")
```

```python
e1 = exp.generate_counterfactuals(x_train[0:1], total_CFs=2, desired_class="opposite")
e1.visualize_as_dataframe(show_only_changes=True)
```

```
Query instance (original outcome : 0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

```
Diverse Counterfactual set (new outcome: 1.0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	67.0	-	Masters	-	-	Other	-	-	1
1	66.0	-	Prof-school	-	-	Other	-	-	1

```
# Restricting age to be between [20,30] and Education to be either {'Doctorate', 'Prof-school'}.
e3 = exp.generate_counterfactuals(x_train[0:1],
                                  total_CFs=2,
                                  desired_class="opposite",
                                  permitted_range={'age':[20,30], 'education':['Doctorate', 'Prof-school']})
e3.visualize_as_dataframe(show_only_changes=True)
```

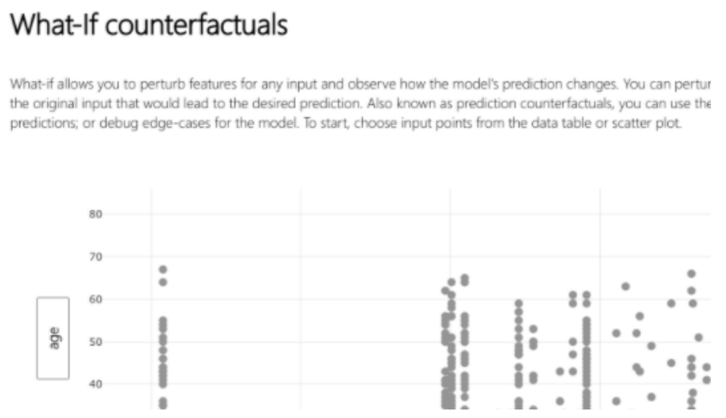
```
Query instance (original outcome : 0)
```

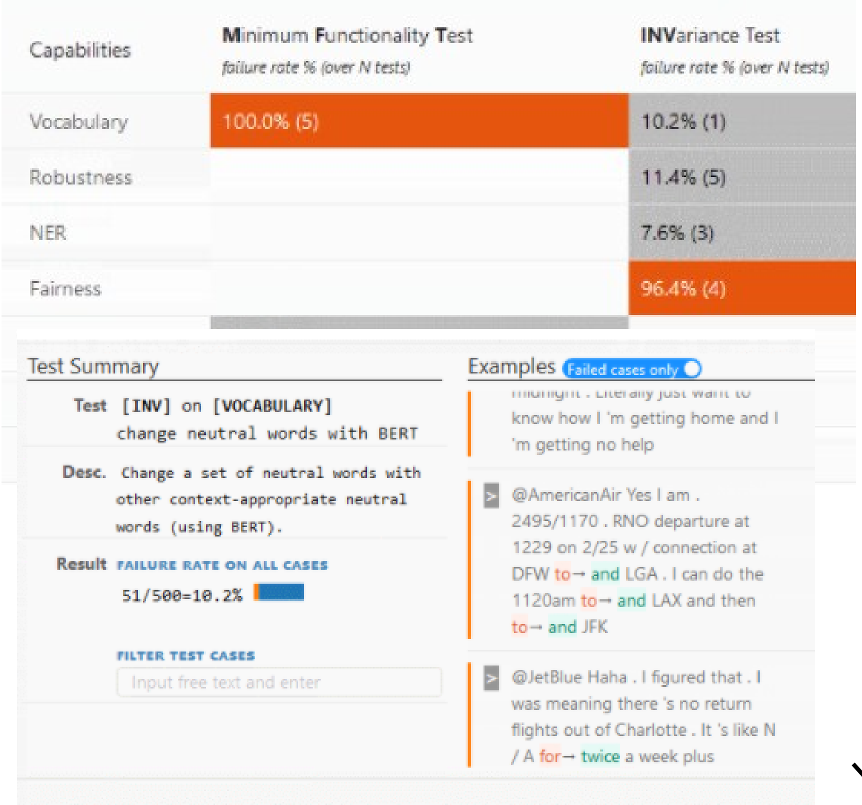
	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

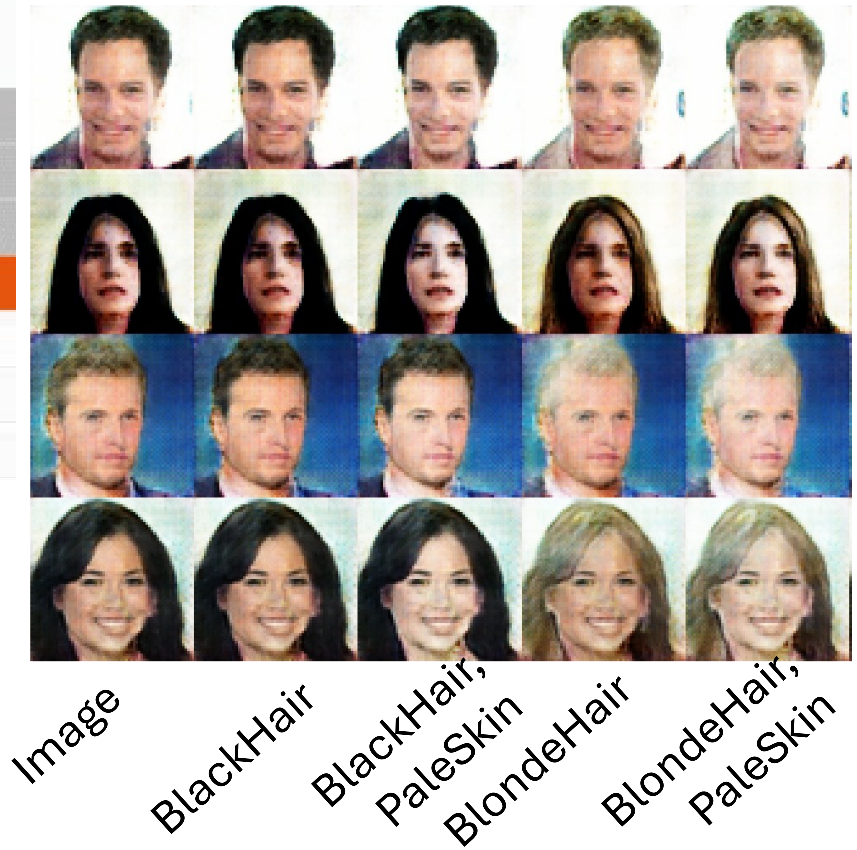
```
Diverse Counterfactual set (new outcome: 1.0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	28.0	Self-Employed	Doctorate	-	Professional	-	Female	21.0	1
1	27.0	Self-Employed	Doctorate	-	Professional	-	Female	50.0	1

Microsoft | Responsible AI







TABULAR DATA

- Keeping other variables constant, does change in gender change model's output?
- Given an input, what is the minimum change in features that changes the model's output?

github.com/interpretml/dice

LANGUAGE DATA

- Keeping other features constant, does change in irrelevant features (e.g., gender) change model's output?
- Mature research

github.com/marcotcr/checklist/
github.com/Microsoft/litmus

IMAGE DATA

- Keeping other features constant, does change in skin color change AI model's output?
- Early-stage research

Conclusion

- Counterfactual explanations offer both interpretability and fidelity
  - Based on intuitive “individual” effects for each example
- Practical, easy to implement
  - DiCE, open-source library for counterfactual model explanations
- Exciting research in generating “realistic” model counterfactuals
  - Need user interfaces to summarize CFs

Collaborators: Chenhao Tan, Divyat Mahajan, Ramaravind Mothilal, Saloni Dash, Vineeth Balasubramanian, Soundarya Krishnan.

To summarize,

Causal ML:

Machine learning + causality

A necessary ingredient for general-purpose AI

- Effect inference (DoWhy)
- ML model attribution (DiCE)

thank you!

Amit Sharma
(@amt_shrma)