Open in app ↗

Search Write

Member-only story

Causal Inference

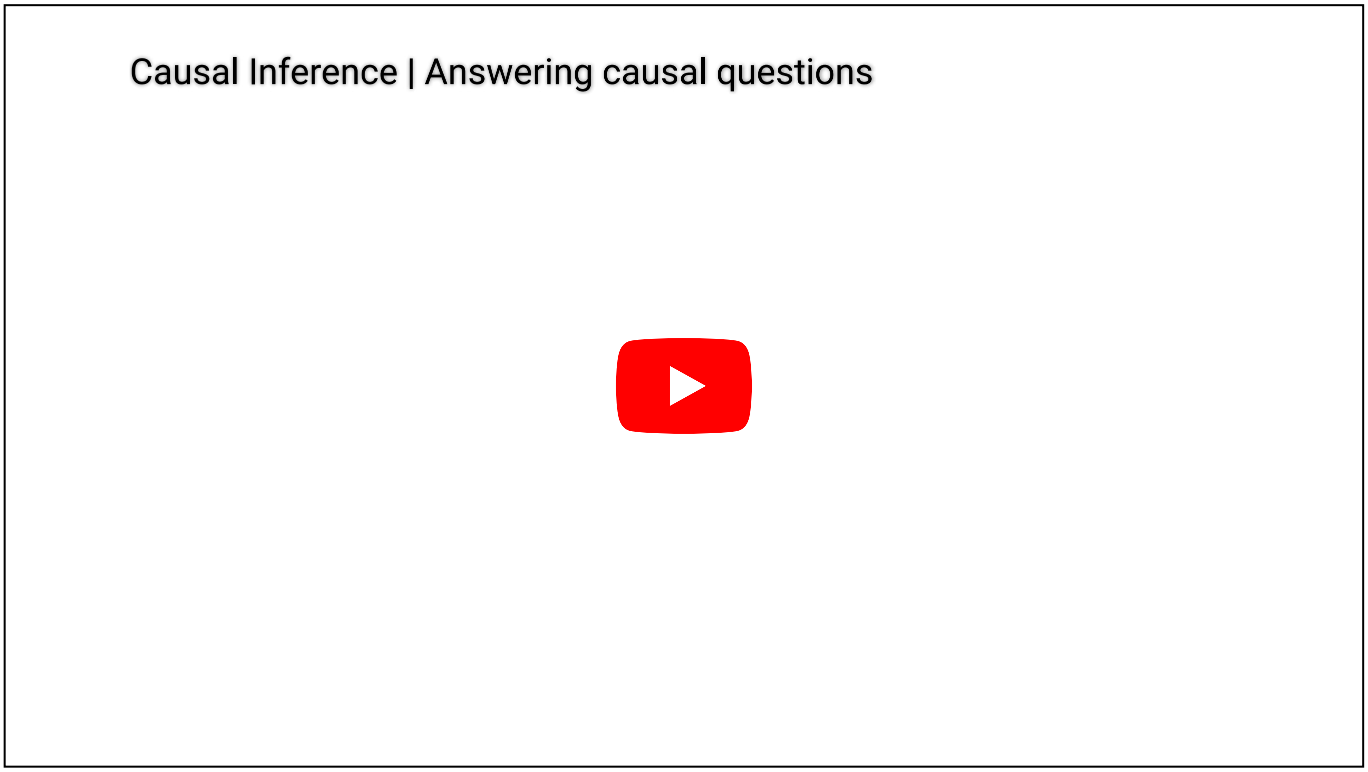
Answering causal questions with Python

Shaw Talebi

Published in Towards Data Science · 9 min read · Oct 15, 2021

510 6

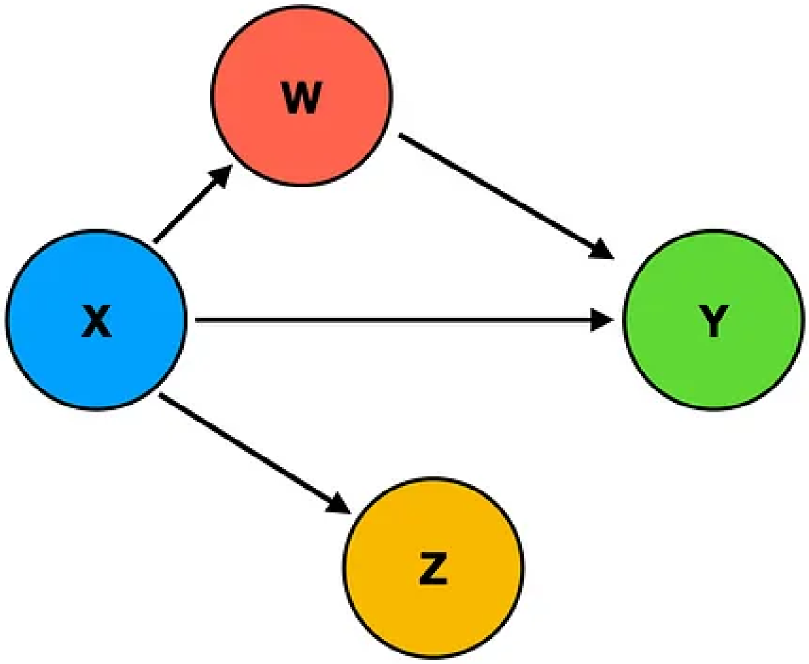
This is the second post in a series of three on causality. In the last post, I introduced this "new science of cause and effect" [1] and gave a flavor for causal inference and causal discovery. In this post, we will dive further into some details of causal inference and finish with a concrete example in Python.



Where to start?

In the last post, I discussed how causality can be represented mathematically via **Structural Causal Models (SCMs)**. SCMs consist of two parts: a graph, which visualizes causal connections, and equations, which express the details of the connections.

To recap, a graph is a mathematical construction consisting **of vertices (nodes) and edges (links)**. Here, I will use the terms graph and network interchangeably. SCMs use a special kind of graph called a **Directed Acyclic Graph (DAG)**, for which all edges are directed and no cycles exist. DAGs are a common starting place for causal inference.

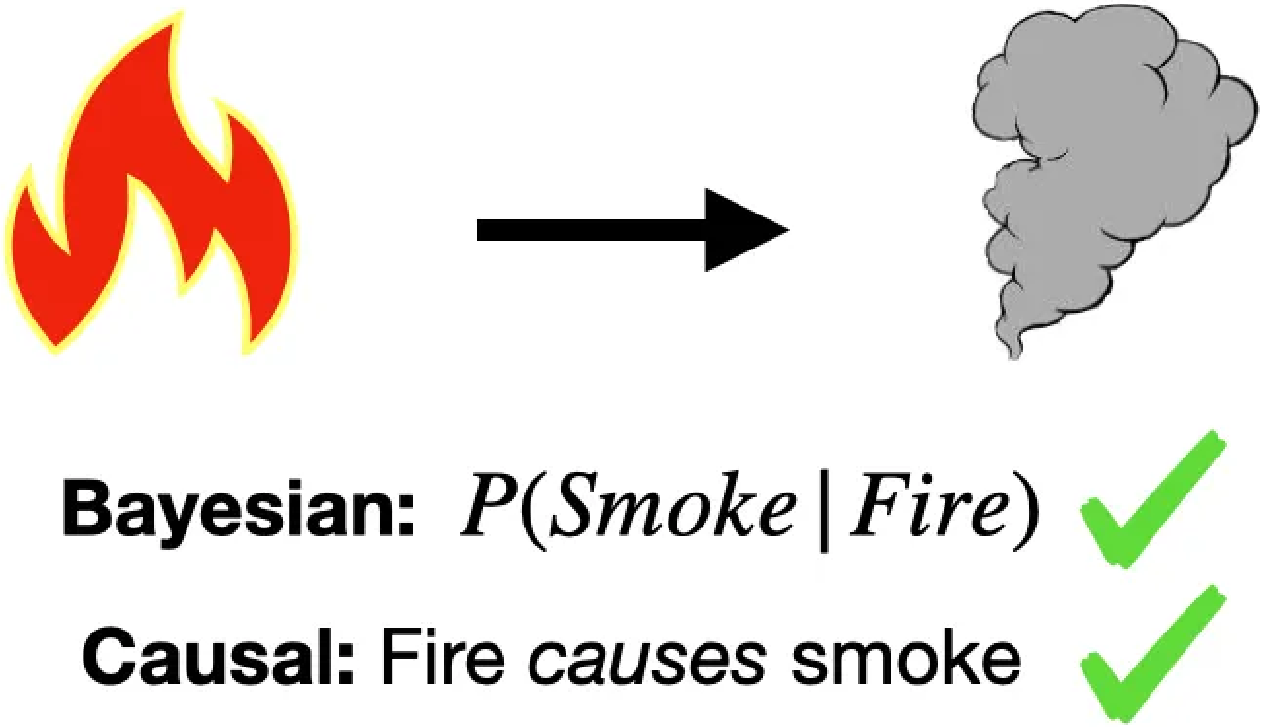


An example causal network. Image by author.

Bayesian vs Causal Networks

An ambiguity for me when first exploring this subject was the difference between **Bayesian networks** and **causal networks**. So I will briefly mention the difference. The enlightened reader can feel free to skip this section.

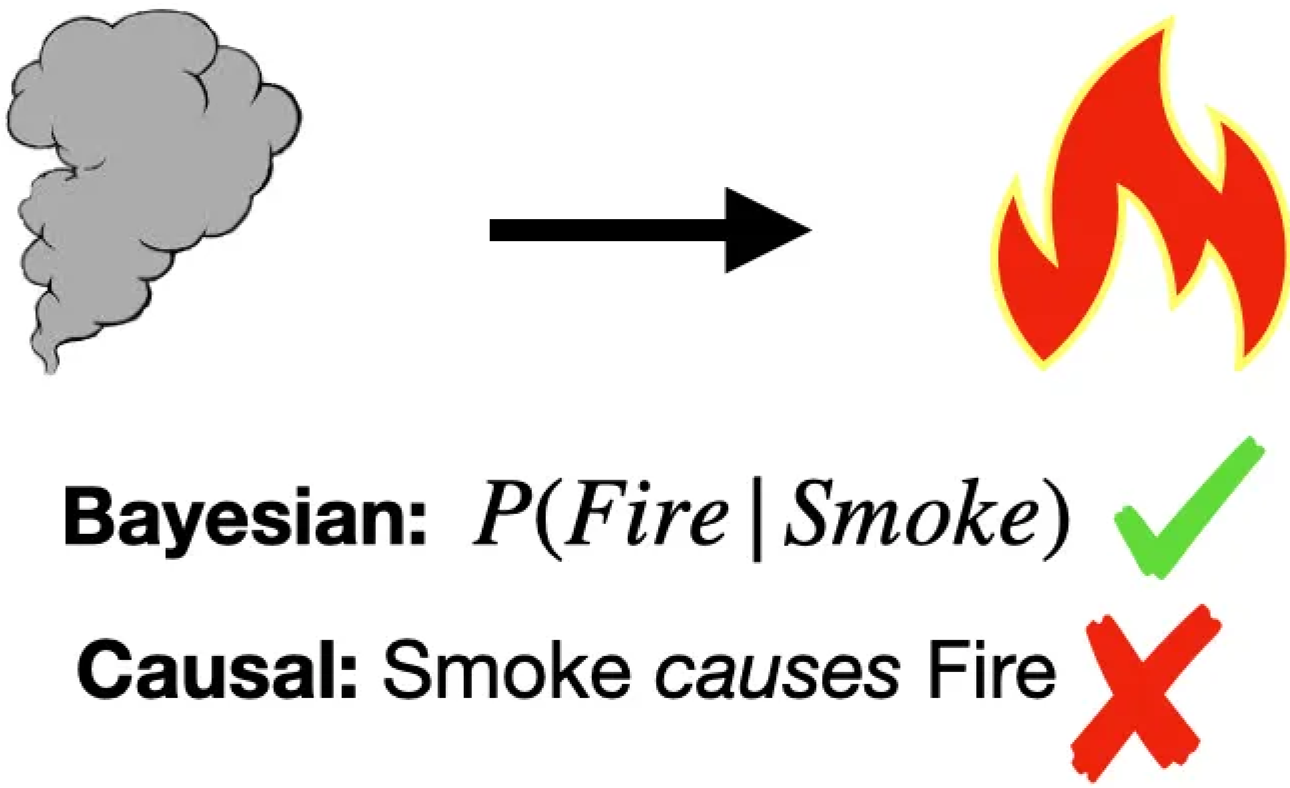
On the surface, Bayesian and causal networks are completely identical. However, the difference lies in their interpretations. Consider the example in the figure below.



Example network that can be interpreted as both Bayesian and causal. Fire and smoke example adopted from Pearl [1]. Image by author.

Here, we have a network with 2 nodes (fire icon and smoke icon) and 1 edge (arrow pointing from fire to smoke). This network can be both a Bayesian or causal network.

The key distinction, however, is when interpreting this network. For a **Bayesian** network, we view the **nodes as variables** and the **arrow as a conditional probability**, namely the probability of smoke given information about fire. When interpreting this as a **causal** network, we still view **nodes as variables**; however, the **arrow indicates a causal connection**. In this case, both interpretations are valid. However, if we were to flip the edge direction, the causal network interpretation would be invalid, since smoke does not *cause* fire.



Example network that can be interpreted as Bayesian, but not causal. Fire and smoke example adopted from Pearl [1]. Image by author.

What is Causal Inference?

Causal inference aims at answering causal questions as opposed to just statistical ones. There are countless applications of causal inference. Answering any of the questions below falls under the umbrella of causal inference.

- Did the treatment directly help those who took it?
- Was the marketing campaign that increased sales this month or the holiday?
- How big of an effect would increased wages have on productivity?

These significant and practical questions may not be easily answered using more traditional approaches (e.g., linear regression or standard machine learning). I aim to illustrate how causal inference can help answer these questions through what I will call the *3 gifts of causal inference*.

3 Gifts of Causal Inference

Gift 1: The do-operator

In the last post, I defined causality in terms of interventions. Omitting some technicalities, it was said that X causes Y if an intervention in X results in a change in Y, while an intervention in Y does not necessarily result in a change in X. Interventions are easy to understand in the real world (like when your friend's candy habit gets out of control), however, *how does that fit into causality's mathematical representation?*

Enter the do-operator. The **do-operator** is a **mathematical representation of a physical intervention**. If we start with the model Z → X → Y, we can simulate an intervention in X by deleting all the incoming arrows to X, and manually setting X to some value x_0.

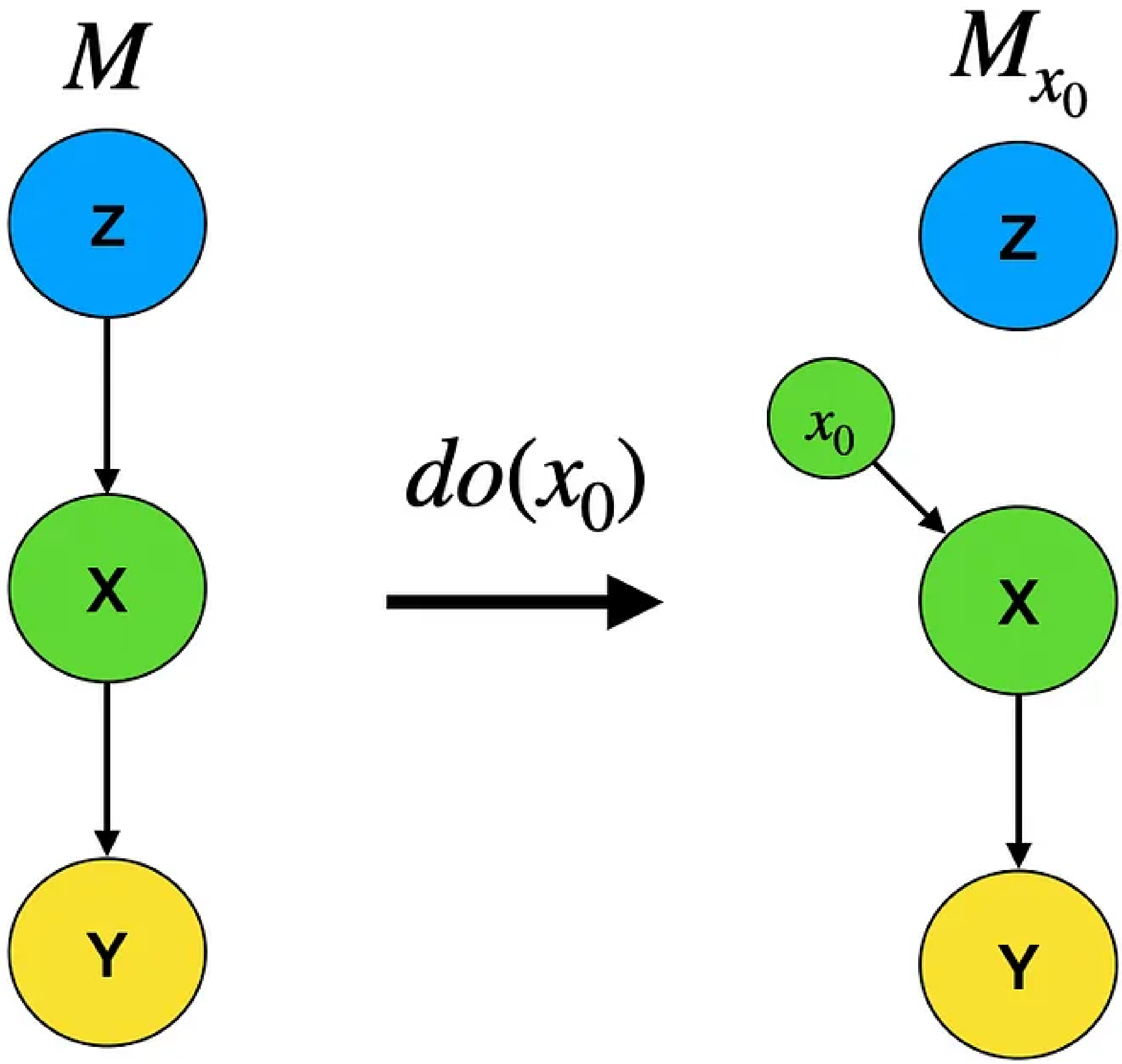


Illustration of how the do-operator works. Image by author.

The power of the do-operator is that it **allows us to simulate experiments**, given we know the details of the causal connections. For example, suppose we want to ask, w*ill increasing the marketing budget boost sales?* If armed with a causal model that includes marketing spend and sales, we can simulate w*hat would happen* if we were to increase marketing spend, and assess whether the change in sales (if any) is worth it. In other words, we can evaluate the **causal effect** of marketing on sales. More on causal effects later.

A major contribution of Pearl and colleagues are the rules of **do-calculus.** This is a *complete* **set of rules that outline how to use the do-operator**.

Notably, do-calculus can translate **interventional distributions** (i.e. probabilities with the do-operator) into **observational distributions** (i.e. probabilities without the do-operator). This can be seen by rules 2 and 3 in the figure below.

Rules of Do-Calculus:

1. Insertion/deletion of observations

$$P(Y \mid do(X), Z, W) = P(Y \mid do(X), Z)$$

If W is irrelevant to Y

2. Action/observation exchange

$$P(Y \mid do(X), Z) = P(Y \mid X, Z)$$

If Z blocks all back-door paths from X to Y

3. Insertion/deletion of actions

$$P(Y \mid do(X)) = P(Y)$$

If there is no causal path from X to Y

Rules of Do-Calculus. Rules are taken from the lecture by Pearl [2]. Image by author.

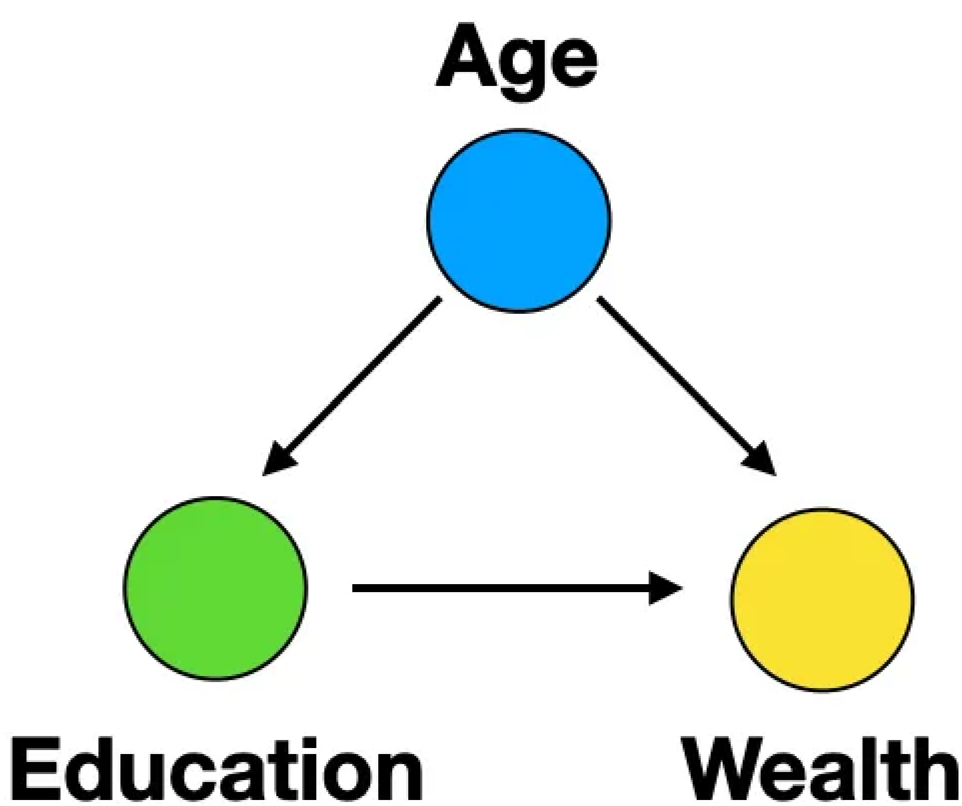
Notice the notation. **P(Y|X)** is the conditional probability that we are all familiar with, that is, the **probability of Y given an observation of X**. While, **P(Y|do(X))** is the **probability of Y given an** ***intervention*** **in X**.

The do-operator is a key tool in the causal inference toolbox. In fact, the next 2 gifts rely on the do-operator.

Causal Effects via the Do-operator
Translating observations into interventions
towardsdatascience.com

Gift 2: Deconfounding Confounding

Confounding is a notion thrown around in statistics. Although I didn't call it by name, this appeared in the previous post via **Simpson's paradox.** A simple example of confounding is shown in the figure below.



A simple example of confounding. Age is a confounder of education and wealth. Image by author.

In this example, age is a confounder of education and wealth. In other words, if trying to evaluate the impact of education on wealth one would need to *adjust* for age. **Adjusting for** (or conditioning on) **age** means that when looking at age, education, and wealth data, one would compare data points **within** age groups, **not between** age groups.

If age were **not** adjusted for, it would not be clear whether education is a true *cause* of wealth or just a *correlate* of wealth. In other words, you couldn't tell whether education directly affects wealth, or just has a *common cause* with it.

For simple examples, confounding is pretty straightforward when looking at a DAG. For 3 variables, the confounder is the variable that points to 2 other variables. *But what about more complicated problems?*

This is where the do-operator provides clarity. Pearl uses the do-operator to define confounding in a clear-cut way. He states **confounding** is **anything that leads to P(Y|X) being different than P(Y|do(X))** [1].

Gift 3: Estimating Causal Effects

This final gift is the main attraction of causal inference. In life, we not only ask ourselves *why*, but *how much?* Estimating causal effects boils down to answering this 2nd question.

Consider graduate school. It is one thing to know that people with graduate degrees make (mostly) more money than those without graduate degrees, but a natural question is, *how much of that is attributable to their degree?* In other words, *what is the treatment effect of a graduate degree on income?*

I will use answering this question as an opportunity to work through a concrete example of using Python to do causal inference.

Causal Effects — Introduction

What is a treatment effect and how to compute it?

towardsdatascience.com

Example: Estimating Treatment Effect of Grad School on Income

In this example, we will use the Microsoft DoWhy library for causal inference [3]. The goal is to estimate the causal effect of a graduate degree on making more than $50k annually. Data is obtained from the UCI machine learning repository [4]. Example code and data can be found at the GitHub repo.

It is important to stress the starting point of all causal inference is a causal model. Here we assume income has only two causes: age and education, where age also is a cause of education. Clearly, this simple model may be missing other important factors. We will investigate alternative models in the next post on **causal discovery**. For now, however, we will focus on this simplified case.

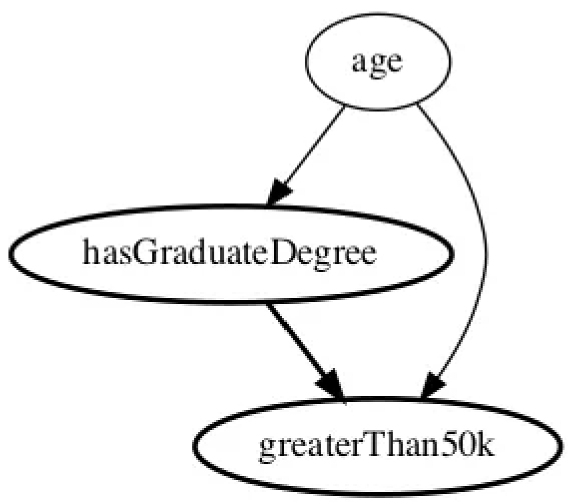
First, we load libraries and data. If you do not have the libraries check out the requirements.txt in the repo.

```
# Import libraries
import pickle
import matplotlib.pyplot as plt

import econml
import dowhy
from dowhy import CausalModel
```

```
# Load Data
df = pickle.load( open( "df_causal_inference.p", "rb" ) )
```

Again the first step is defining our causal model i.e. DAG. DoWhy makes it easy to create and view models.

```
# Define causal model
model=CausalModel(data = df,
                  treatment= "hasGraduateDegree",
                  outcome= "greaterThan50k",
                  common_causes="age",
                  )

# View model
model.view_model()
from IPython.display import Image, display
display(Image(filename="causal_model.png"))
```



Our (assumed) causal model. Image by author.

Next, we need an estimand. This is basically a recipe that gives us our desired causal effect. In other words, it tells us how to compute the effect of education on income.

```python
# Generate estimand
identified_estimand=
model.identify_effect(proceed_when_unidentifiable=True)
print(identified_estimand)
```

```
Estimand type: nonparametric-ate

### Estimand : 1
Estimand name: backdoor
Estimand expression:
         d
────────────────────(Expectation(greaterThan50k|age))
d[hasGraduateDegree]
Estimand assumption 1, Unconfoundedness: If U→{hasGraduateDegree} and U→greaterThan50k then P(greaterThan50k|hasGraduateDegre
e,age,U) = P(greaterThan50k|hasGraduateDegree,age)

### Estimand : 2
Estimand name: iv
No such variable found!

### Estimand : 3
Estimand name: frontdoor
No such variable found!
```

Output of estimand generation. Image by author.

Finally, we compute the causal effect based on the estimand. Here we use a meta-learner [5] from the EconML library, which estimates conditional average treatment effects for discrete targets.

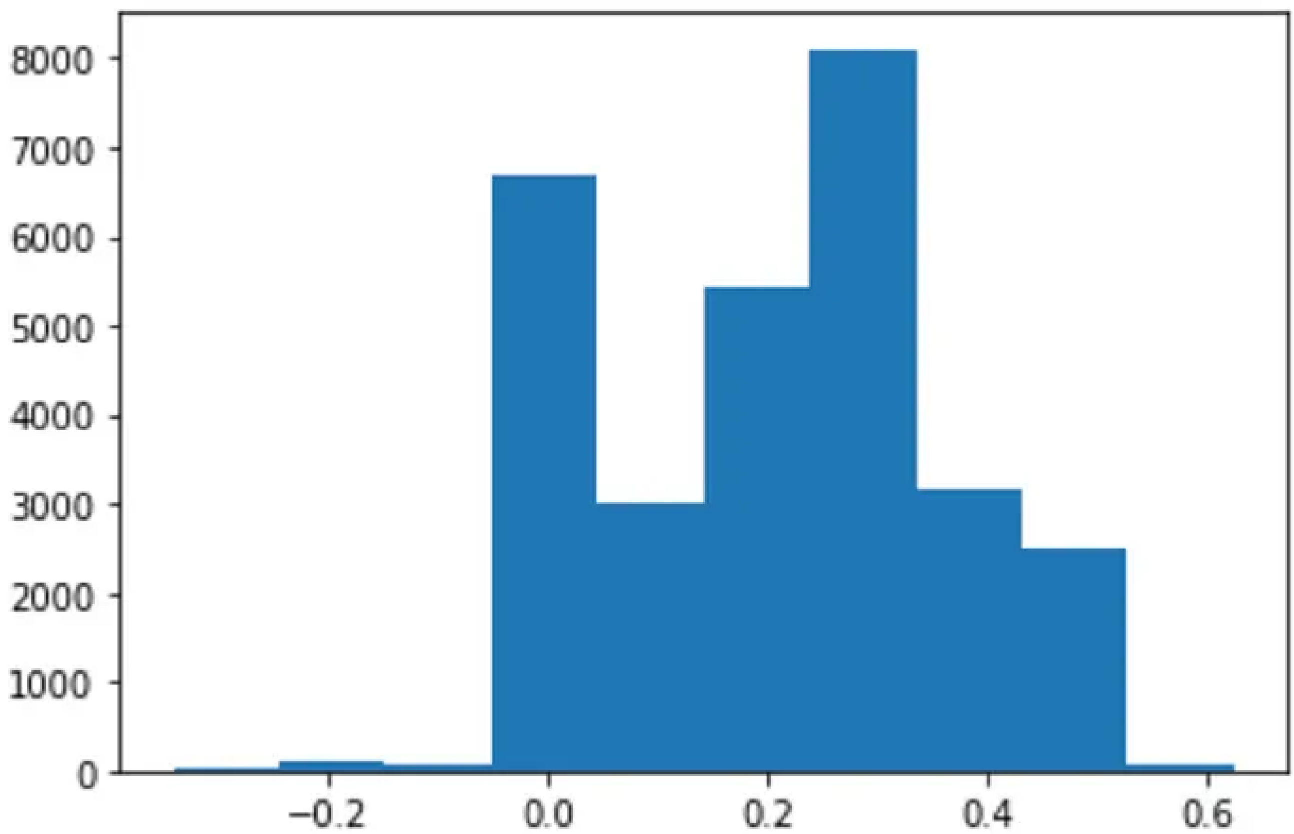
```python
# Compute causal effect using metalearner
identified_estimand_experiment =
model.identify_effect(proceed_when_unidentifiable=True)

from sklearn.ensemble import RandomForestRegressor
metalearner_estimate =
model.estimate_effect(identified_estimand_experiment,
method_name="backdoor.econml.metalearners.TLearner",
confidence_intervals=False,
method_params={
     "init_params":{'models': RandomForestRegressor()},
     "fit_params":{}
              })

print(metalearner_estimate)
```

```
*** Causal Estimate ***

## Identified estimand
Estimand type: nonparametric-ate

### Estimand : 1
Estimand name: backdoor
Estimand expression:
         d
────────────────────(Expectation(greaterThan50k|age))
d[hasGraduateDegree]
Estimand assumption 1, Unconfoundedness: If U→{hasGraduateDegree} and U→greaterThan50k then P(greaterThan50k|hasGraduateDegre
e,age,U) = P(greaterThan50k|hasGraduateDegree,age)

## Realized estimand
b: greaterThan50k~hasGraduateDegree+age
Target units: ate

## Estimate
Mean value: 0.20340255646236685
Effect estimates: [ 0.31961958  0.20493831  0.35577517 ...  0.15907199 -0.01266913
  0.19505072]
```

Output of causal estimation. Image by author.

The average causal effect is about 0.20. This can be interpreted as, having a graduate degree increases your probability of making more than $50k annually by 20%. Noting this is the average effect, it is important to consider the full distribution of values to assess whether the average is representative.

```
# Print histogram of causal effects
plt.hist(metalearner_estimate.cate_estimates)
```



Distribution of causal effects. Image by author.

The figure above shows the distribution of causal effects across samples. Clearly, the distribution is *not* Gaussian. Which tells us the mean is not representative of the overall distribution. Further analysis diving into cohorts based on causal effects may help uncover actionable information about "who" benefits most from a graduate degree.

Regardless, solely basing a decision to go to grad school on potential income, may be an indication you don't really want to go to grad school. 🤷🏽‍♀️

Causal Effects via Regression

3 Popular Techniques With Python Example Code

towardsdatascience.com

Conclusion

Causal inference is a powerful tool for answering natural questions that more traditional approaches may not resolve. Here I sketched some big ideas from causal inference and worked through a concrete example with code. As stated before, a causal model is the starting point for all causal inference. Usually, however, we don't have a good causal model in hand. This is where **causal discovery** can be helpful, which is the topic of the next post.

👉 **More on Causality:** Causal Effects Overview | Causality: Intro | Causal Inference | Causal Discovery

Resources

Connect: My website | Book a call | Ask me anything

Socials: YouTube 🎥 | LinkedIn | Twitter

Support: Buy me a coffee ☕️

The Data Entrepreneurs

A community for entrepreneurs in the data space. 👉 Join the Discord!

medium.com

[1] The Book of Why: The New Science of Cause and Effect by Judea Pearl

[2] Pearl, J. (2012). The Do-Calculus Revisited. arXiv:1210.4852 **[cs.AI]**

[3] Amit Sharma, Emre Kiciman. DoWhy: An End-to-End Library for Causal Inference. 2020. https://arxiv.org/abs/2011.04216

[4] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml/datasets/census+income

[5] Künzel, Sören R., et al. "Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning." *Proceedings of the National Academy of Sciences*, vol. 116, no. 10, Mar. 2019, pp. 4156–65. *www.pnas.org*, https://doi.org/10.1073/pnas.1804597116.

Causality | Causal Inference | Machine Learning | Python | Confounding