

Causal Explanations



inFERENCe

posts on machine learning,
statistics, opinions on things
I'm reading in the space



[Home](#)

May 24, 2018

ML beyond Curve Fitting: An Intro to Causal Inference and do-Calculus

You might have come across [Judea Pearl's new book](#), and a [related interview](#) which was widely shared in my social bubble. In the interview, Pearl dismisses most of what we do in ML as curve fitting. While I believe that's an overstatement (conveniently ignores RL for example), it's a nice reminder that most productive debates are often triggered by controversial or outright arrogant comments. Calling machine learning alchemy was a great recent example. After reading the article, I decided to look into his famous do-calculus and the topic causal inference once *again*.

Again, because this happened to me semi-periodically. I first learned do-calculus in a (very unpopular but advanced) undergraduate course Bayesian networks. Since then, I have re-encountered it every 2-3 years in various contexts, but somehow it never really struck a chord. I always just thought "this stuff is difficult and/or impractical" and eventually forgot about it and moved on. I never realized how fundamental this stuff was, until now.

This time around, I think I fully grasped the significance of causal reasoning and I turned into a full-on believer. I know I'm late to the game but I almost think it's basic hygiene for people working with data and conditional probabilities to understand the basics of this toolkit, and I feel embarrassed for completely ignoring this throughout my career.

In this post I'll try to explain the basics, and convince you why you should think about this, too. If you work on deep learning, that's an even better reason to understand this. Pearl's comments may be unhelpful if interpreted as contrasting deep learning with causal inference. Rather, you should interpret it as highlighting causal inference as a huge, relatively underexplored, application of deep learning. Don't get discouraged by causal diagrams looking a lot like Bayesian networks (not a coincidence seeing they were both pioneered by Pearl) they don't compete with, they complement deep learning.

Basics

Let's say we have i.i.d. data sampled from some joint $p(x,y,z,\dots)$.

Say we are ultimately interested in how variable y behaves given x . At a high level, one can ask this question in two ways:

- **observational** $p(y|x)$:

What is the distribution of Y given that I observe variable X takes value x .

- **interventional** $p(y|\text{do}(x))$:

What is the distribution of Y if I were to set the value of X to x .

This describes the distribution of Y I would observe if I intervened in the data generating process by artificially forcing the variable X to take value x , but otherwise simulating the rest of the variables according to the original process that generated the data. Note that the **data generating procedure** is NOT the same as the joint distribution $p(x,y,z,\dots)$ and this is an important detail.

What exactly is $p(y|do(x))$?

is the joint distribution of data which we would observe if we actually carried out the intervention in question.

$$P_{do(X=x)}(x, y, z, \dots)$$

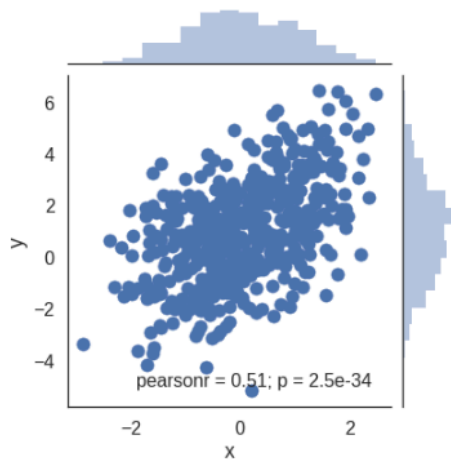
$$p(y \mid do(X = x))$$

is the conditional distribution we would learn from data collected in randomized controlled trials or A/B tests where the experimenter controls x .

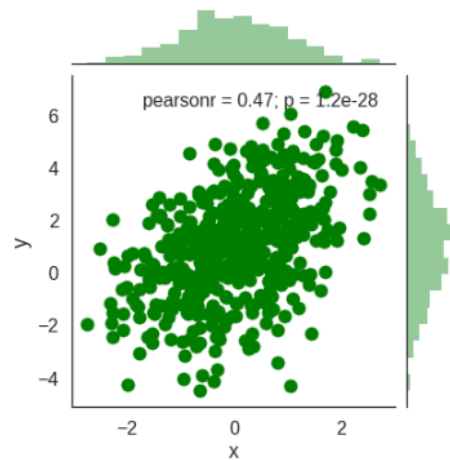
What is an intervention?

Based on the joint distribution the three scripts are indistinguishable.

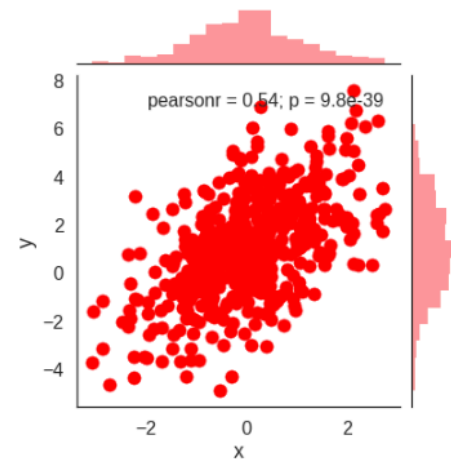
```
x = randn()
y = x + 1 + sqrt(3)*randn()
```



```
y = 1 + 2*randn()
x = (y-1)/4 + sqrt(3)*randn()/2
```

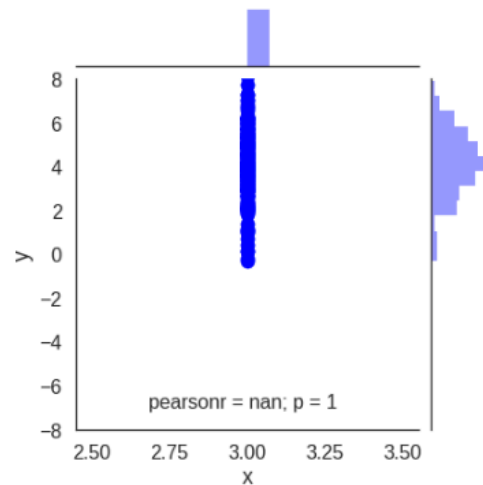


```
z = randn()
y = z + 1 + sqrt(3)*randn()
x = z
```

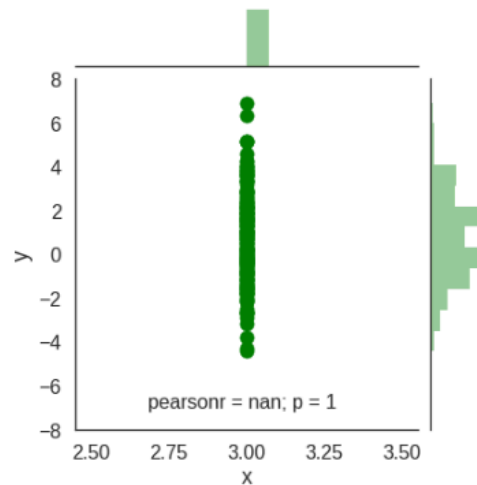


What is an intervention?

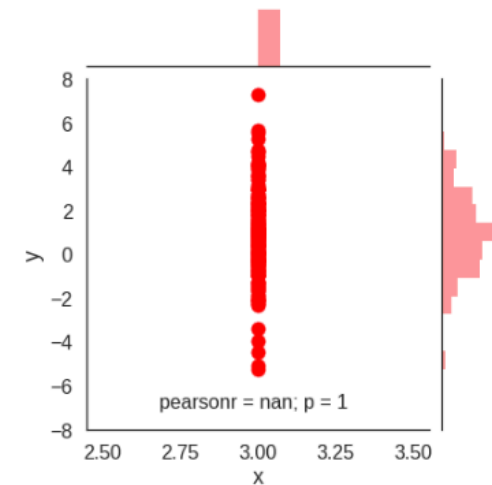
```
x = randn()  
x = 3  
y = x + 1 + sqrt(3)*randn()  
x = 3
```



```
y = 1 + 2*randn()  
x = 3  
x = (y-1)/4 + sqrt(3)*randn()/2  
x = 3
```

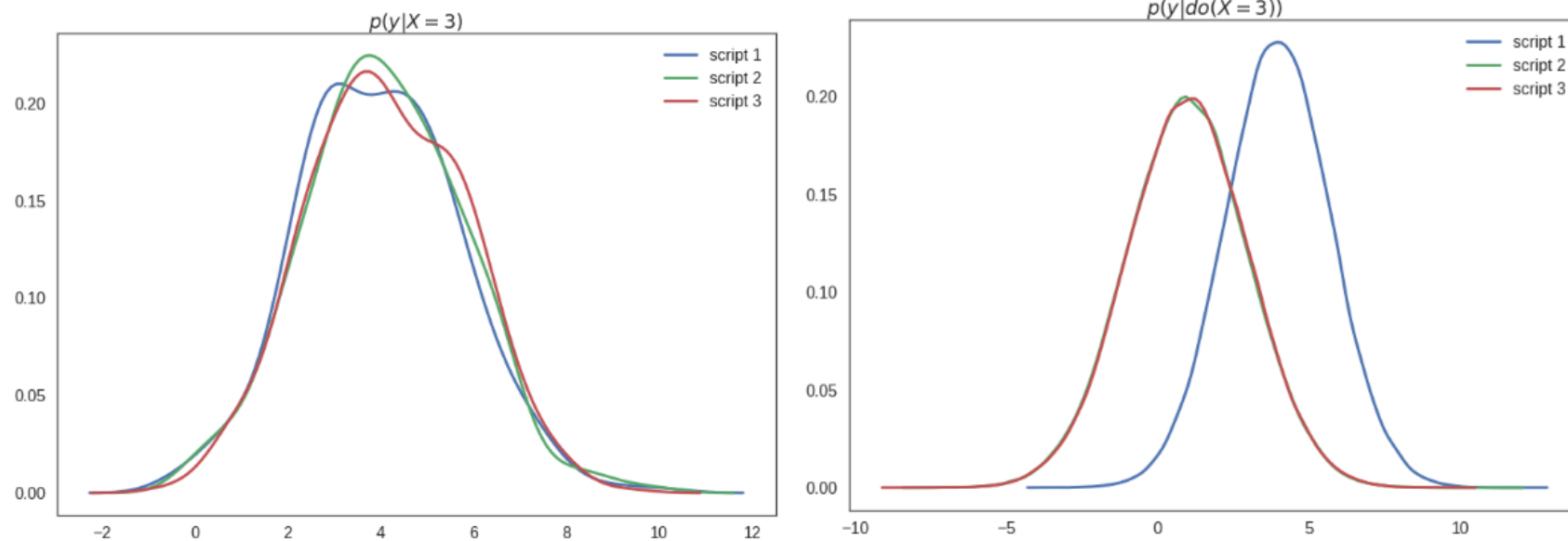


```
z = randn()  
x = 3  
x = z  
x = 3  
y = z + 1 + sqrt(3)*randn()  
x = 3
```



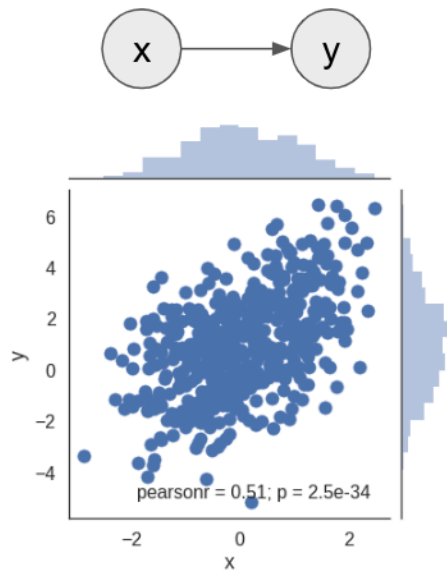
What is an intervention?

Here is a better look at the marginal distribution of y under the intervention. The scripts behave differently under intervention. The joint distribution of data alone is insufficient to predict behavior under interventions.

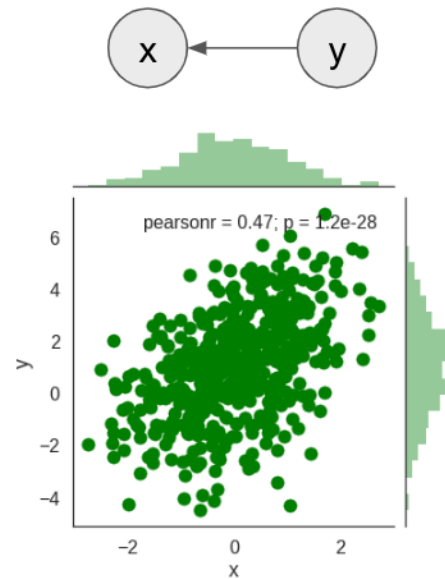


What is an intervention?

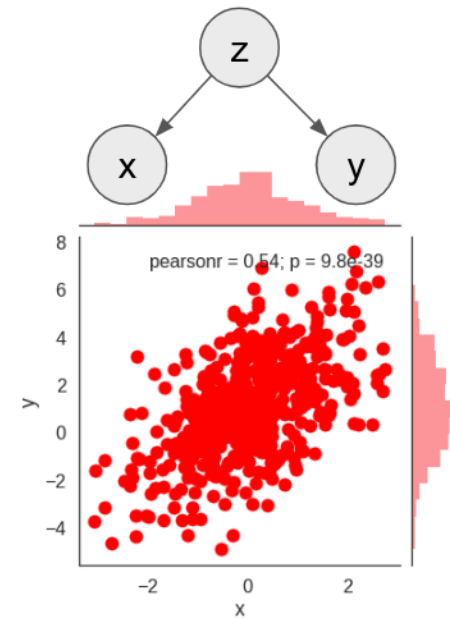
```
x = randn()  
y = x + 1 + sqrt(3)*randn()
```



```
y = 1 + 2*randn()  
x = (y-1)/4 + sqrt(3)*randn()/2
```



```
z = randn()  
y = z + 1 + sqrt(3)*randn()  
x = z
```

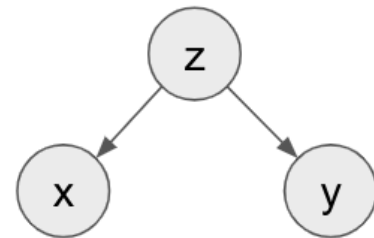




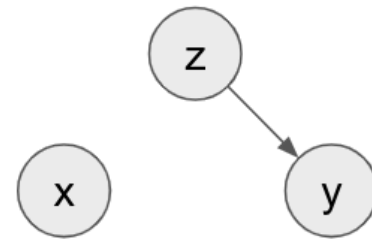
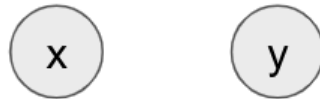
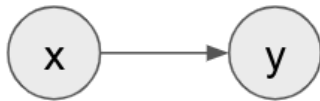
$$P(y|do(X)) = p(y|x)$$



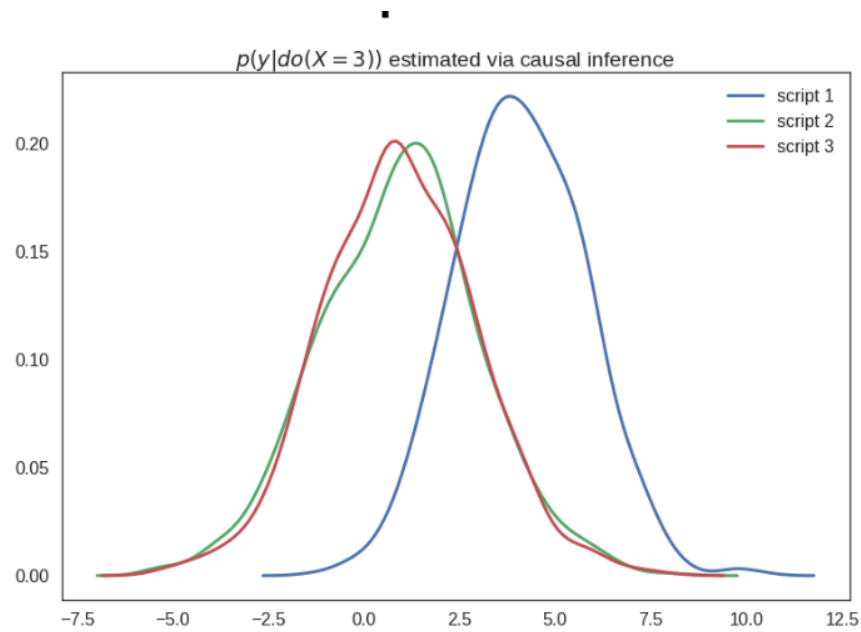
$$P(y|do(X)) = p(y)$$



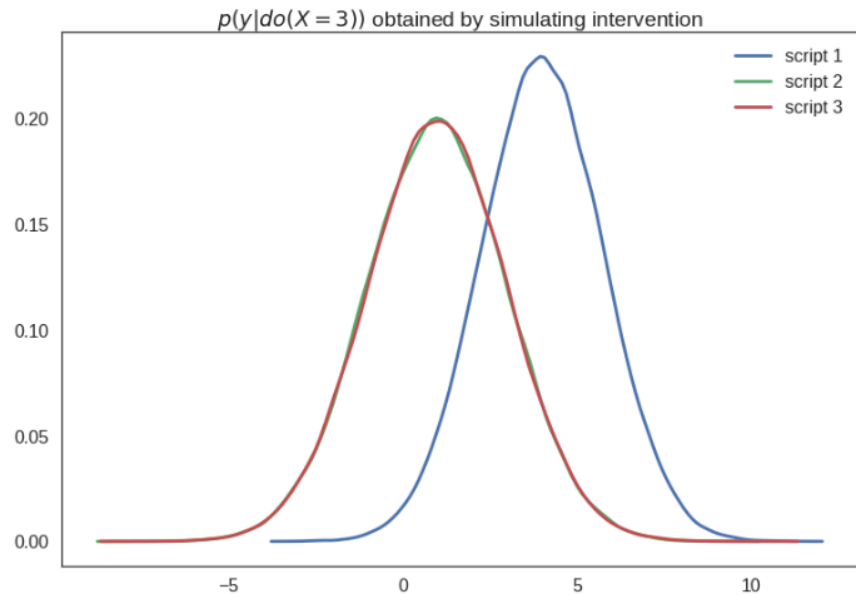
$$P(y|do(X)) = p(y)$$



What is an intervention?



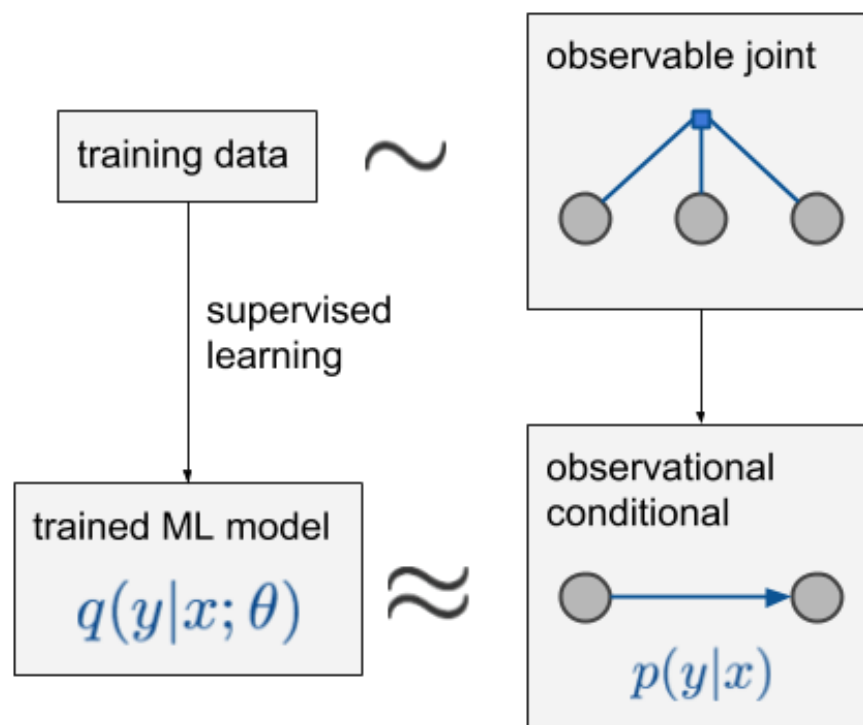
estimated from joint + causal diagram



actually running the experiment

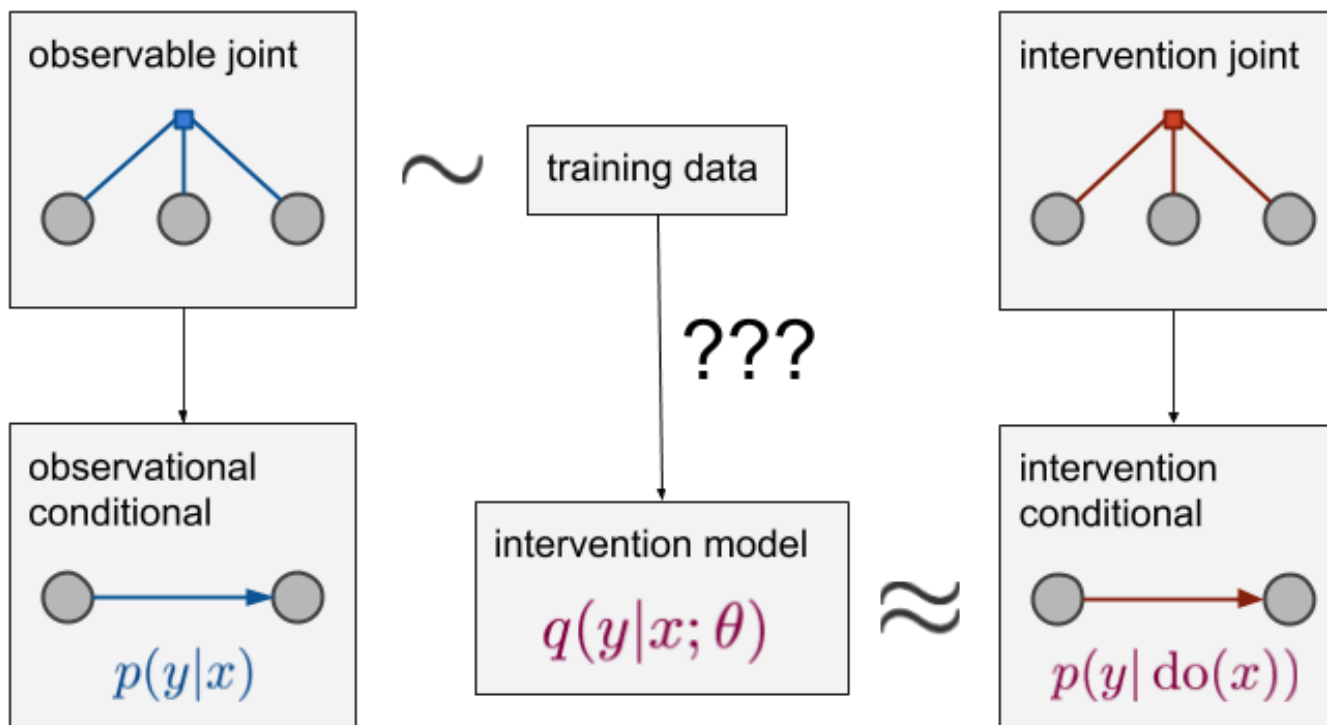
The supervised learning case:

We are interested in predicting y from x , and say that z is a third variable which we do not want to infer but we can also measure. We only care about $p(y|x)$.



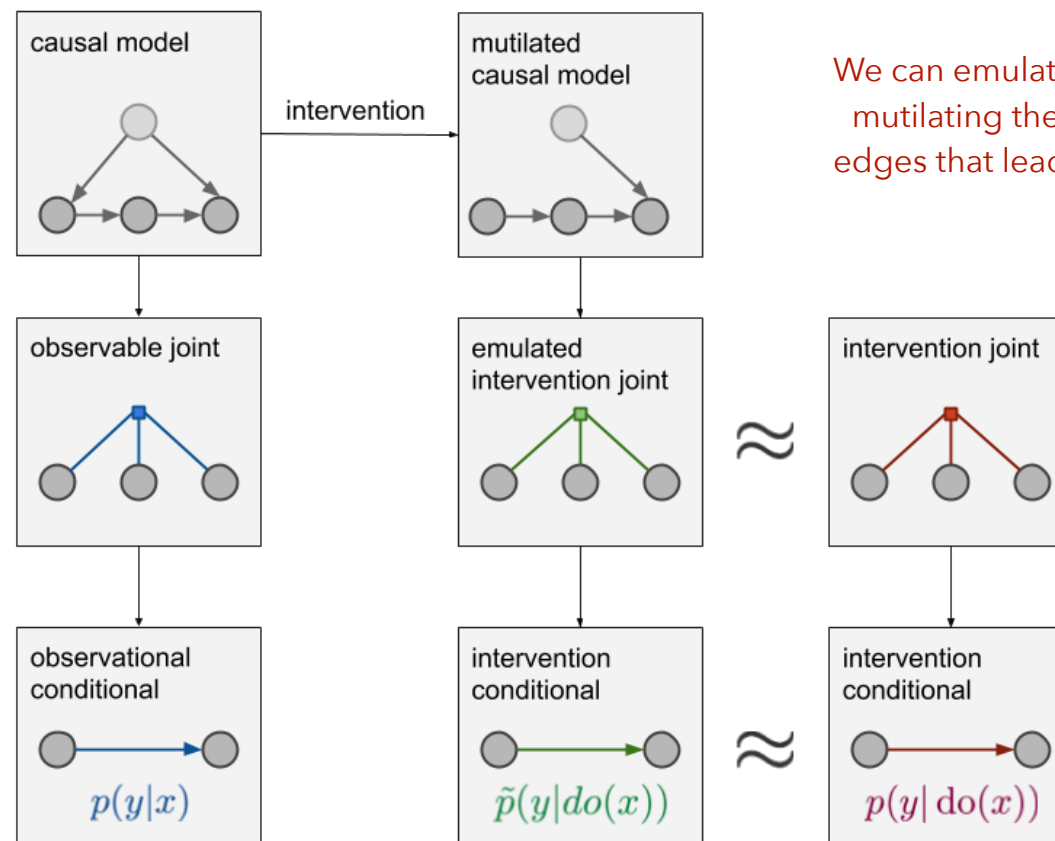
The intervention model:

Now, what if we're actually interested in $p(y|\text{do}(x))$ rather than $p(y|x)$? This is what it looks like:



The causal model:

If we want to establish a connection between the blue and the red joints, *we must* introduce additional assumptions about the causal structure of the data generating mechanism.

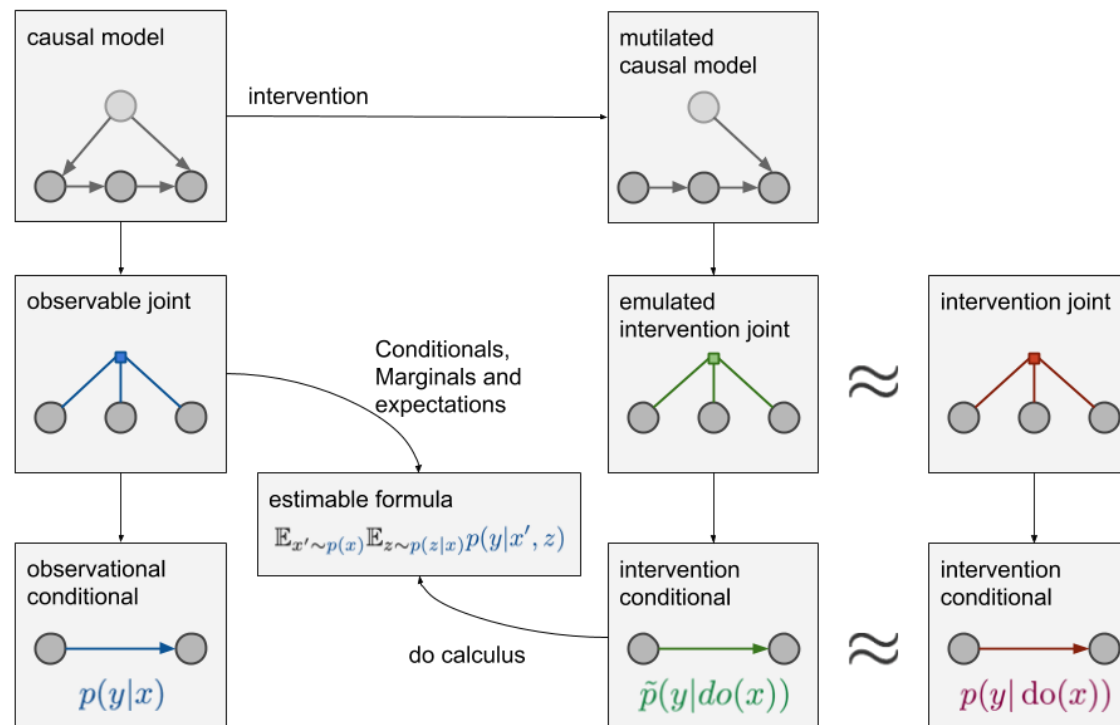


We can emulate the effect of intervention by mutilating the causal network: deleting all edges that lead into nodes in a do operator.

Do calculus:

Now the question is, **how can we say anything about the green conditional when we only have data from the blue distribution.** We are in a better situation than before as we have the causal model relating the two. To cut a long story short, this is what the so-called *do-calculus* is for.

Do-calculus allows us to massage the green conditional distribution until we can express it in terms of various marginals, conditionals and expectations under the blue distribution.



Counterfactuals: David Blei's election example

Given that Hilary Clinton did not win the 2016 presidential election, and given that she did not visit Michigan 3 days before the election, and given everything else we know about the circumstances of the election, what can we say about the probability of Hilary Clinton winning the election, had she visited Michigan 3 days before the election?

Let's try to unpack this. We are are interested in the probability that:

- she *hypothetically* wins the election

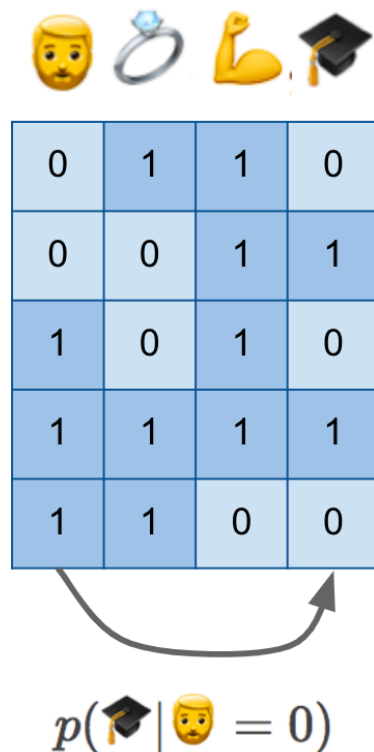
conditioned on four sets of things:

- she lost the election
- she did not visit Michigan
- any other relevant an observable facts
- she *hypothetically* visits Michigan

Why would quantifying this probability be useful? Mainly for credit assignment.

Queries

Let's start with the simplest thing one can do to attempt to answer a question: collect some data about individuals, whether they have beards, whether they have PhDs, whether they are married, whether they are fit, etc. Here's a cartoon illustration of such dataset:



The table contains 5 rows and 4 columns of data. Above the table are four emojis: a man with a beard, a diamond ring, a flexing arm, and a graduation cap. A curved arrow points from the bottom-left cell (1, 1) to the bottom-right cell (5, 4).

0	1	1	0
0	0	1	1
1	0	1	0
1	1	1	1
1	1	0	0

$p(\text{graduation cap} | \text{beard} = 0)$

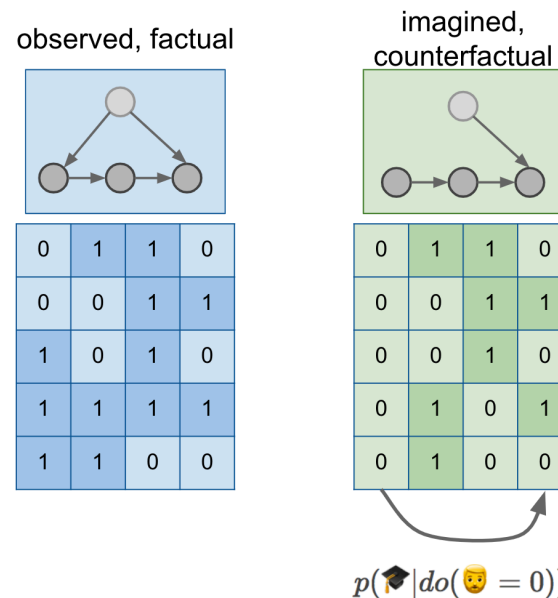
We can make **observational queries** to this dataset: *What is the conditional probability of possessing a PhD degree given the absence of a beard.*

Queries

This is not an interventional query.
This is a counterfactual query!

Let's now suppose that I want an answer to this question: *Given that I have a beard, and that I have a PhD degree, and everything else we know about me, with what probability would I have obtained a PhD degree, had I never grown a beard?*

Let's consider the scenario for **interventional queries**:

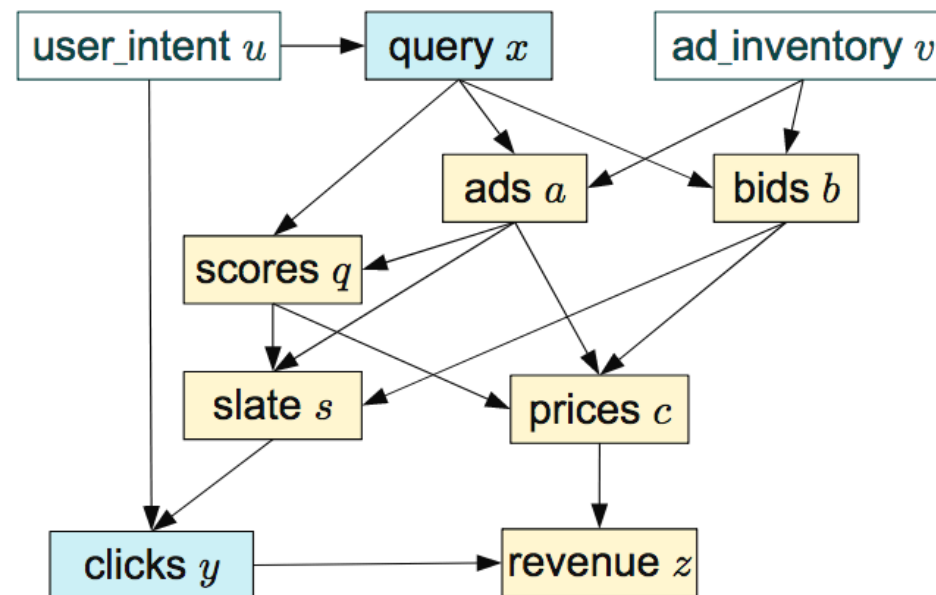


Can $p(\text{PhD} | do(\text{Beard} = 0))$ express the counterfactual probability we seek?

Counterfactual queries

$p(\text{🎓} | do(\text{👤}=0))$ talks about a **randomly sampled individual**, while a **counterfactual** talks about a **specific individual**!

To get an answer to our question we have to step beyond causal graphs and introduce another concept: **structural equation models**.



Counterfactual queries

The dependencies shown by the diagram are equivalently encoded by the following set of equations:

x	$=$	$f_1(u, \epsilon_1)$	Query context x from user intent u .
a	$=$	$f_2(x, v, \epsilon_2)$	Eligible ads (a_i) from query x and inventory v .
b	$=$	$f_3(x, v, \epsilon_3)$	Corresponding bids (b_i).
q	$=$	$f_4(x, a, \epsilon_4)$	Scores ($q_{i,p}, R_p$) from query x and ads a .
s	$=$	$f_5(a, q, b, \epsilon_5)$	Ad slate s from eligible ads a , scores q and bids b .
c	$=$	$f_6(a, q, b, \epsilon_6)$	Corresponding click prices c .
y	$=$	$f_7(s, u, \epsilon_7)$	User clicks y from ad slate s and user intent u .
z	$=$	$f_8(y, c, \epsilon_8)$	Revenue z from clicks y and prices c .

For each node in the graph above we now have a corresponding function f_i . The arguments of each function are the causal parents of the variable it instantiates, e.g. f_1 computes x from its causal parent u , and f_2 computes a from its causal parents x and v . In order to allow for nondeterministic relationship between the variables, we additionally allow each function f_i to take another input, ϵ_i which you can think of as a random number.

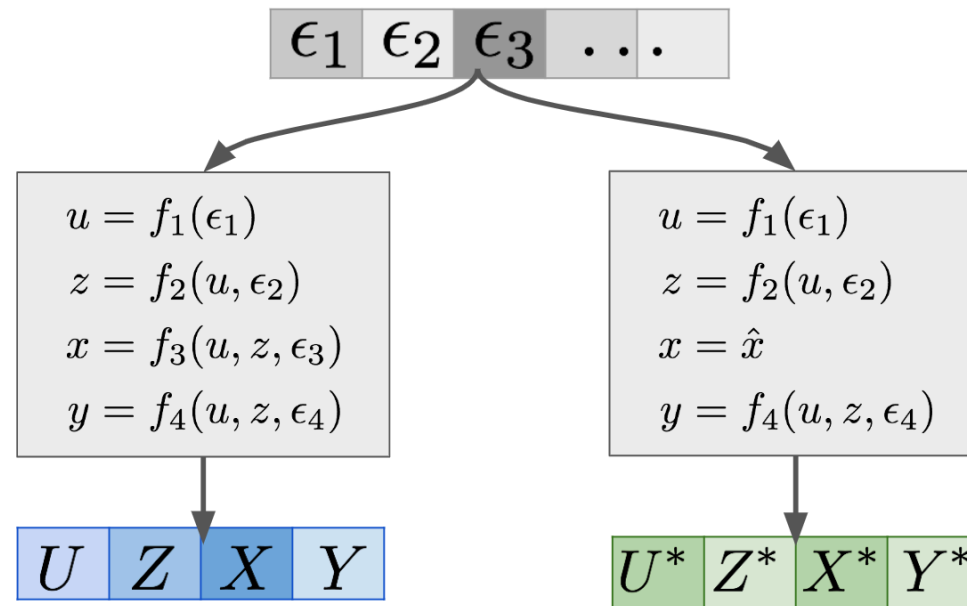
Counterfactual queries

The structural equation model (SEM) entails the causal graph, in that you can reconstruct the causal graph by looking at the inputs of each function. It also entails the joint distribution, in that **you can "sample" from an SEM by evaluating the functions in order, plugging in the random ϵ s where needed.**

In a SEM an **intervention** on a variable, say q , can be modeled by deleting the corresponding function, f_4 , and replacing it with another function.

For example $\text{do}(Q=q_0)$ would correspond to a simple assignment to a constant $f_4(x,a)=q_0$.

Counterfactual queries



A SEM is essentially a generative model of data, which uses some noise variables $\epsilon_1, \epsilon_2, \dots$ and turns them into observations (U, Z, X, Y) in this example. This is shown in the left-hand branch of the graph above. Now if you want to make counterfactual statements under the intervention $X = \hat{x}$, you can construct a *mutilated* SEM, which is the same SEM except with f_3 deleted and replaced with the constant assignment $x = \hat{x}$. This modified SEM is shown in the right-hand branch. If you feed the ϵ s into the mutilated SEM, you get another set of variables (U^*, Z^*, X^*, Y^*) , shown in green. These are the features of the twin as it were. This joint generative model over (U, Z, X, Y) and (U^*, Z^*, X^*, Y^*) defines a joint distribution over the combined set of variables $(U, Z, X, Y, U^*, Z^*, X^*, Y^*)$. Therefore, now you can calculate all sorts of conditionals and marginals of this joint.

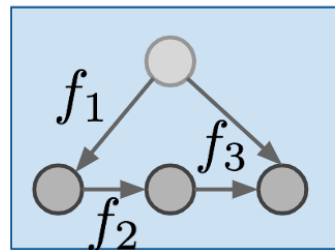
Counterfactual queries

If you feed the first row of epsilons to the blue structural equation model, you get the first blue datapoint 0110. If you feed the same epsilons to the green SEM, you get the first green datapoint (0110). If you feed the second row of epsilons to the models, you get the second rows in the blue and green tables, and so on...

ϵ_1 ϵ_2 ϵ_3

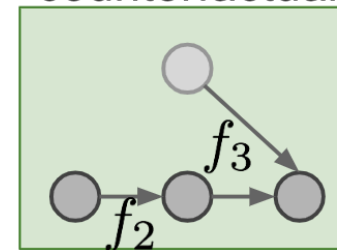
0.1	0.3	0.7	...
0.7	0.1	0.0	...
0.4	0.8	0.6	...
1.0	0.2	1.0	...
0.7	0.3	0.5	...

observed, factual



0	1	1	0
0	0	1	1
1	0	1	0
1	1	1	1
1	1	0	0

imagined,
counterfactual



0	1	1	0
0	0	1	1
0	0	1	0
0	1	0	1
0	1	0	0

$$p(\text{🎓}^* | \text{👤}^* = 0, \text{👤} = 1, \text{💍} = 1, \text{💪} = 1, \text{🎓} = 1)$$

Exercises

The screenshot shows the GitHub interface for the repository `DataScienceUB / ExplainableDataScience`. The repository has 12 commits, 1 branch, 0 releases, and 1 contributor. The main content area displays a list of files with their commit history and timestamps.

Repository: `DataScienceUB / ExplainableDataScience`

Buttons: Watch (1), Star (0), Fork (0)

Navigation: Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, Settings

Repository Name: Explainable Data Science Course

Manage topics

Statistics: 12 commits, 1 branch, 0 releases, 1 contributor

Actions: Branch: master, New pull request, Create new file, Upload files, Find file, Clone or download

File Name	Commit Message	Time Ago
1. ExplainableMachineLearningPermutations.ipynb	Add files via upload	2 minutes ago
2. ExplainableMachineLearningPDP.ipynb	Add files via upload	10 minutes ago
FIFA 2018 Statistics.csv	Add files via upload	11 days ago
README.md	Update README.md	12 days ago
taxi.csv	Add files via upload	11 days ago

<https://github.com/DataScienceUB/ExplainableDataScience>

Exercises

1. **Permutations:** <https://colab.research.google.com/github/DataScienceUB/ExplainableDataScience/blob/master/1.%20ExplainableMachineLearningPermutations.ipynb>
2. **Partial Dependence Plots:** <http://colab.research.google.com/github/DataScienceUB/ExplainableDataScience/blob/master/2.%20ExplainableMachineLearningPDP.ipynb>
3. **SHAP:** <https://colab.research.google.com/github/DataScienceUB/ExplainableDataScience/blob/master/3.%20ExplainableMachineLearningShap.ipynb>
4. **LIME for TEXT:** <https://colab.research.google.com/github/DataScienceUB/ExplainableDataScience/blob/master/4.%20ExplainableMachineLearningText.ipynb>