DoWhy

User Guide > ··· > Asking and Answering What-If Questions > Computing...

Computing Counterfactuals

By computing counterfactuals, we answer the question:

> I observed a certain outcome z for a variable Z where variable X was set to a value x. What would have happened to the value of Z, had I intervened on X to assign it a different value x' ?

As a concrete example, we can imagine the following:

> I'm seeing unhealthy high levels of my cholesterol LDL ($Z = 10$). I didn't take any medication against it in recent months ($x = 0$). What would have happened to my cholesterol LDL level (Z), had I taken a medication dosage of 5g a day ($X := 5$)?

Note that the estimation of counterfactuals based on Pearl's graphical causal model framework requires stronger assumptions than the generation of interventional samples (see the Understanding the method section for more details).

How to use it

To see how we can use this method, let's generate some data:

```
>>> import networkx as nx, numpy as np, pandas as pd
>>> from dowhy import gcm
```

```
>>> X = np.random.uniform(low=0, high=10, size=2000)
>>> Y = -2*X + np.random.normal(loc=0, scale=5, size=2000)
>>> Z = 3*Y + 80 + np.random.normal(loc=0, scale=5, size=2000)
>>> training_data = pd.DataFrame(data=dict(X=X, Y=Y, Z=Z))
```

Skip to main content

If we think of the cholesterol example, one could think of Y here as some kind of body measurement that affects the cholesterol LDL level, such as a derivative of the body's ability to absorb the medication or its metabolism.

Next, we'll model cause-effect relationships and fit the models to the data. Estimating counterfactuals requires an invertible SCM that can be represented by additive noise models:

```
>>> causal_model = gcm.InvertibleStructuralCausalModel(nx.DiGraph([('X', 'Y'), ('Y
>>> gcm.auto.assign_causal_mechanisms(causal_model, training_data)
```

```
>>> gcm.fit(causal_model, training_data)
```

Suppose we have observed the values $x = 0, y = 5, z = 110$. We are interested in knowing if taking medication, e.g., setting $x := 5$, would have led to a smaller value for z. The counterfactual value of z for this scenario can be estimated in the following way:

```
>>> gcm.counterfactual_samples(
>>>     causal_model,
>>>     {'X': lambda x: 5},
>>>     observed_data=pd.DataFrame(data=dict(X=[0], Y=[5], Z=[110])))
   X         Y         Z
0  5 -4.82026  80.62051
```

As we can see, X takes our treatment-/intervention-value of 5, and Y and Z take deterministic values that are *based on our trained causal models* and the fixed observation. For example, if X were 0 and Y were 5, we would expect Z to be 95 based on the data generation process. However, we observed Z to be 110, indicating a noise value of approximately ~15 in this *particular* sample. With knowledge of this hidden noise factor, we can estimate the counterfactual value of Z had we set X to 5, which is approximately ~80 as shown in the result above.

We can also provide these noise values directly to the function:

```
>>> gcm.counterfactual_samples(
>>>     causal_model,
>>>     {'X': lambda x: 5},
>>>     noise_data=pd.DataFrame(data=dict(X=[0], Y=[5], Z=[15])))
   X         Y          Z
0  5 -4.845684  80.431559
```

Skip to main content

As we see, with X set to 5 and $y = -2 \cdot x + 5 = -5$, z should be approximately ~65. But we know the hidden noise for Z is approximately ~15. So the counterfactual outcome is again $z = 3 * y + 80 + 15 = 80$.

It's important to note that the estimated values we obtained are not exact as we see in the results, since the model parameters were not learned perfectly. Specifically, the learned coefficients for Y and Z are not precisely -2 and 3, respectively, as suggested by the data generation process. As usual in machine learning, more data or fine-tuning models can help to improve accuracy.

Related example notebooks

- Counterfactual Analysis in a Medical Case

Understanding the method

Counterfactuals in graphical causal models are very similar to Simulating the Impact of Interventions, with an important difference: when performing interventions, we look into the future, for counterfactuals we look into an alternative past. To reflect this in the computation, when performing interventions, we first recreate all noise values for a specific observation and then estimate the counterfactual outcome. However, this requires stronger modeling assumptions than generating interventional samples.

Recall that the causal mechanism of a node Y is represented as $Y := f(X, N)$, where X are the parents of Y and N is the noise. To generate interventional samples for Y, we can take any random value from N, but for counterfactuals, we need to first reconstruct the specific noise value (based on the model) that led to our observation. This requires the causal mechanism to be invertible with respect to N, although it does not need to be invertible with respect to X. A common modeling assumptions that ensures this are additive noise models of the form $Y := f(X) + N$, where the noise can be reconstructed by $N = Y - f(X)$. Note that currently only continuous data is supported for counterfactual estimation, while there is no restriction on the data type for generating interventional samples.

To further clarify the role of the noise here, let's revisit the example in the introduction about high levels of cholesterol LDL. Seeing that there are many unobserved factors that can impact cholesterol levels, such as exercise and genetics, the question arises: If I had taken medication,

Skip to main content

the subject specific unobserved factors (i.e., noise) constant and only change the amount of hypothetically taken medicine. In practice this can be achieved by first reconstructing the noise and then use that specific value to estimate the LDL levels after intervening on the amount of medicine. Here it is crucial to use the reconstructed noise value rather than randomly sampling it from the noise distribution. Otherwise, one may see a reduction in LDL levels, not because of the medication itself, but because of the generated noise value that coincidentally causes low levels. Assuming the modeling assumptions are approximately correct, we can then analyze whether the medication would have helped in the counterfactual scenario.

Note

Remark on invertible mechanisms: Generally, mechanisms that are invertible with respect to N allow us to estimate point counterfactuals. However, it is also possible to allow some mechanisms to be non-invertible. In this case, however, we would obtain a counterfactual *distribution* based on the observational evidence, which may not necessarily be point-wise.

Previous
‹ Simulating the Impact of

Next
Predicting outcome for out-of- ›

Created using Sphinx 7.1.2.

Built with the PyData Sphinx Theme 0.14.4.