User Guide > Modeling Causal Relations > Refuting a...

Refuting a Causal Graph

Now that we have obtained a causal graph for our problem, a natural question is: Is the causal graph correct? In other words, is the graph consistent with the available data?

To answer this question, we use the fact that each causal graph entails a set of conditional independence statements over its nodes. These conditional independencies are called Local Markov Conditions (LMCs). If the dataset does not satisfy any of the LMCs implied by the graph, then the graph is invalid.

To understand local markov conditions, let us consider a system with three nodes: X, Y and Z. These nodes can be arranged in three graph structures:

1. Chain, Z→X→Y
2. Fork, Z←X→Y
3. Collider, Z→X←Y

The first two graphs entail that Z and Y should be independent given X. And the third graph entails that Z and Y are independent, but they will become dependent when conditioned on X. To understand how these conditions are derived, you can read up on d-separation.

Given a candidate graph and a dataset, we can use these conditions to refute a given graph. For example, if a user provides the Chain graph for a dataset, we can check whether $Z \perp\!\!\!\perp Y | X$. If not, then we can conclude that the user-provided graph is not a valid DAG for the dataset. However, if the dataset satisfies the constraint, it does not mean that the user-provided graph is valid. It is plausible that the true graph may be the Fork graph, which also implies the same conditional independence. Therefore, the test may be able to refute some incorrect graphs, but it cannot uniquely determine the correct graph for a dataset in general. Continuing with our example, if the user finds that their Chain graph is invalid, they may edit the graph and propose a Collider graph. The conditional independence constraint in the refutation test will now fail to invalidate the graph and one can go ahead with downstream analysis.

Skip to main content

For larger graphs, the number of conditional independence tests can be a big list. DoWhy can enumerate and run those tests automatically. See also the example notebook Falsification of User-Given Directed Acyclic Graphs. Next, we show how to test a single independence constraint and then test the validity of the full graph.

‹ Previous
Learning causal structure from data

Next ›
Performing independence tests

Created using Sphinx 7.1.2.

Built with the PyData Sphinx Theme 0.14.4.