Investigating Pearl's Theory of Counterfactuals

Sophie Song
Supervised by David Poole

Cognitive Systems and Computer Science
University of British Columbia

Abstract

Representing counterfactual queries as Bayesian networks using Judea Pearl's twin model allows us to evaluate these queries probabilistically. We provide an example to demonstrate a causally dependent but probabilistically independent relationship has belief updates that do not reflect our intuition. Additionally, we also show that the parents of the observed variable in the counterfactual portion of the twin model do not need to be represented. We aim to demonstrate how our decision in representation of the world requires certain types of knowledge, such as causal relations, which cannot be inferred solely through numerical values or interventional methods.

Introduction

A counterfactual sentence generally takes the form of:

> If A had been true, B would have been true.

where A and B were actually both observed as being false and we are specifying a hypothetical case where A is contrary to what was observed and inquiring/asserting the result.

For example, we might ask "if I studied harder for my exam, would I have passed?". In asking this question, we presuppose that I had not studied hard for my exam and I had already failed the exam. We then ask what would happen in a contrary scenario, given what had already happened.

Another example commonly used to demonstrate counterfactuals is "if Lee-Harvey Oswald had not shot John F. Kennedy, then Kennedy would have still been alive", which presupposes that we know that Oswald shot Kennedy and that Kennedy is dead.

These statements are difficult to evaluate because they cannot be tested like other inquiries. While you can test a drug's efficacy through multiple trials with controlled variables (via intervention), you cannot go back in time and and run through multiple situations involving Oswald and Kennedy and how many times Oswald shot Kennedy and how many times Kennedy dies to obtain a probability distribution.

Pearl investigated proposing counterfactual queries in this form to evaluate them:

> If A' were true, what is the probability that C would have been true, given that we know A is false?

In this query, A represents the actual observations we made, A' represents the hypothetical scenario with the alternative observation, C represents the outcome where A happens, and C' represents the outcome where we know B but A happens instead. By asking a counterfactual query in this way, we can evaluate them using probabilistic tools like Bayesian networks. This is what Pearl proposes – we represent counterfactual arguments using Bayesian networks. More specifically, Alexander Balke and Judea Pearl propose that we use a 'twin model' to represent the counterfactual example. [2] One part of the model represents the real-world observations and the other part of the model represents the counterfactual observations. These parts should be identical in structure and have the same noise variables affecting each pair of variables.

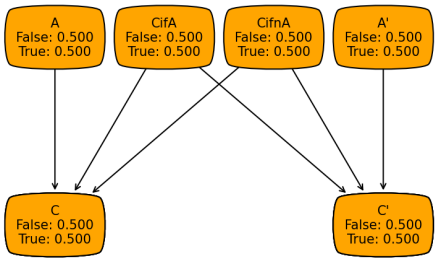

Figure 1: Simple twin model example with 2 variables in a causal chain and two noise variables for the child node

More formally, Pearl proposes that we use a set of variables describing the world $X = \{X_1, X_2, ..., X_n\}$ which may have corresponding observations $x_1, x_2, ..., x_n$. Pearl uses an asterisk to denote the counterfactual variables with an asterisk (e.g. $\hat{x}$) but we will be refer to them as the primed version (e.g. Figure 1, $A \,\&\, A'$) or with (CF) affixed at the end (e.g. the figures later in this paper).

Thus, representing the counterfactual query in notation, it would generally be $P(C \mid A', A)$ and more specifically $P(C \mid A' = True, A = False)$.

To evaluate this using value iteration,

$$P(C \mid A' = T, A = F) = \alpha \sum_{C,CifA,CifnA,C'} \Big[f_1(C, CifA, CifnA) \times f_2(C', CifA, CifnA) f_3(CifA) f_4(CifnA) \Big] \quad (1)$$

Where α is some normalization constant obtained at the end of our calculation. Approximate methods could also be used to obtain this probability distribution in a complicated Bayesian network.

It is important to note that Bayesian networks represent *causal relationships* between variables. This should be intuitive since a counterfactual statement would not make sense without a causal relationship. For example, we might say:

> If it had not rained, the grass would not be wet.

But, if we swapped the antecedent and consequent, the resulting sentence would be a bit odd. It would look like this:

> If the grass was not wet, it would not have rained.

This feels strange because wet grass does not cause rain, so we do not expect rain to change or be affected by the grass being wet or not. But, the first sentence feels right because we expect rain to affect the grass being wet. Asserting that it had rained leads to a likely consequence of the grass being wet because we know that rain causes grass to become wet. So, the first sentence matches our intuition because we would expect a contrary outcome as a result of a contrary observation because the event observed causes the outcome. But, the second sentence does not match our intuition because we do not expect a contrary outcome when given a contrary observation because the event is not caused by the outcome.

Bayesian networks are a useful tool because they can model causal relationships and the usage of probabilities in the network, helping us both represent the actual causal relationship of real-world events based on what we know about the world and also use probabilities to discuss what would be likely given this information. They also allow us to update our probabilities based on evidence, allowing us to query on counterfactuals with observed variables.

Most of the literature involving counterfactual arguments and Pearl's twin model representation of them in Bayesian networks focus primarily on use cases and tend to be quite complicated. Instead, we decided to use a simple working example to explore certain properties, such as the necessity of certain nodes and the noise function/variables chosen, which will be discussed in this report.

Our Working Example

Let us suppose that there are two individuals: Ann and Bill. Ann lives in downtown Vancouver and Bill lives on the campus of University of British Columbia. They agreed to go on a date in downtown Vancouver. Bill is preparing for the date when he realizes that if he does not leave right at this moment, he will certainly be late. So, he uses Uber to hail a ride and notices that there are two cars equidistant from him. Let us call them Car 1 and Car 2. Both cars are identical in every way, so he does not care which one he is picked up by (this indifference is represented by the equal probability of being picked up by either car). He then has a certain chance of being on time for the date which is the same in both cars. Based on whether Bill is on time or not, Ann has different chances of going on a second date with him. If Bill is on time she is more likely to go on a second date with him, but if he is late, she's less likely to go on a second date. This is the Bayesian network representation of the scenario:

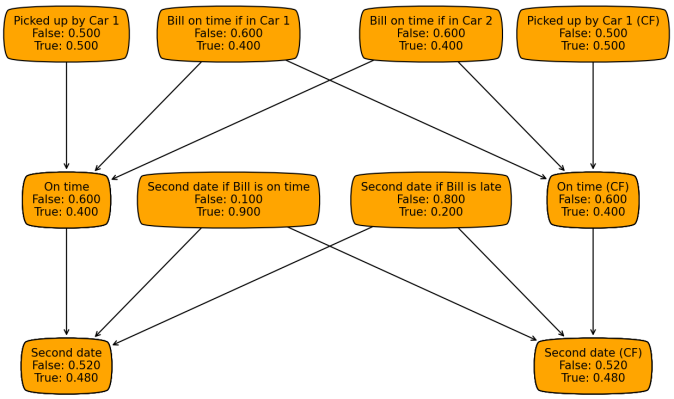

Figure 2: Bayesian network representation of the counterfactual example with no observations

Redundant Observed Parents

If we use Pearl's twin model to represent the counterfactual world, every variable needs to be represented. But, let us consider a situation where we observe Bill being on time and we ask what happens when Bill isn't on time – do we really need to know if Bill was in Car 1 or Car 2?

Here is what the network looks like in this scenario:

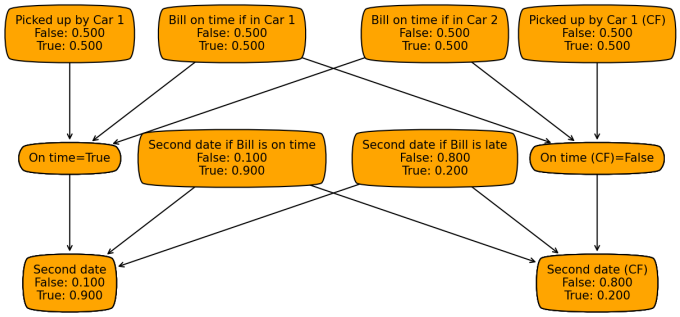

Figure 3: Bayesian network representation of the counterfactual example where Bill is observed to have been on time and we query on if Bill was late.

Is it really necessary to represent the parent nodes of

On Time and its counterfactual twin? Knowing properties of Bayesian Networks, we can use the d-separation rules to show that the lower half of this network is independent from the rest of the network.

There are four scenarios in d-separation:

0. Nodes that are not connected are independent from each other.
1. Knowing a parent node makes the parent/ancestors above the known node independent from the child.
2. Knowing a parent node makes the siblings independent from one another.
3. Not knowing a common child node makes the parents independent from each other.

Since we know On Time and On Time (CF), then using the first rule of d-separation we can show that Picked up by Car 1 is independent from Second Date (and their counterfactual counterparts) since On Time is observed. Similarly, the noise variables, Bill being on time in Car 1 and Bill being on time in Car 2, are also blocked by the evidence from On Time.

So, if we prune the unnecessary parent nodes in this scenario, it would look like this.

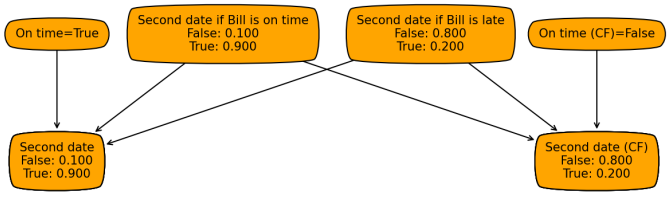

Figure 4: Bayesian network representation of the counterfactual example where Bill is observed to have been on time and we query on if Bill was late, with the parents of On Time pruned.

We can tell that the resulting probabilities are identical in Figure 3 and Figure 4. As shown using d-separation rules, we already knew that the network in Figure 4 is independent from its parents. This example only has one parent which is easy to prove independence using d-separation. While this independence seems to hold for multiple parents with some experiments by running different counterfactual arguments and different observed variables (see the Section 11.5 of Poole and Mackworth's AI textbook [4] for a firing squad counterfactual argument example with pruned parents of the observed variable and the corresponding codebase in AIPython), it is not as simply proven using d-separation rules as the single-parent case. Our goal is not to provide a rigorous proof that this independence holds for all cases, but to demonstrate that it can be proven in single-parent cases. Further investigation needs to be conducted to determine whether a proof can be written for an observed variable with multiple parents.

Counterintuitive Results

In our example with Bill and Ann, Bill being on time is causally dependent (hence the arrows) on Bill being picked up by either Car 1 or Car 2. But, note that Bill's probability of being in either car is independent from being on time. Because Bill is indifferent to being in either car and the probability of Bill being on time in either car are the same, his probability of being on time in general is independent of which car he's in. On the other hand, Bill and Ann going on a second date is causally *and* probabilistically dependent on Bill being on time since we know that Bill being on time or not affects Ann's decision to go on another date with him.

Figure 4 shows the network where we know that Bill is picked up in Car 1 and they agree to go on a second date. Here, we're setting the contrary observation if Bill had been picked up in Car 2 to determine how likely they are to go a second date.

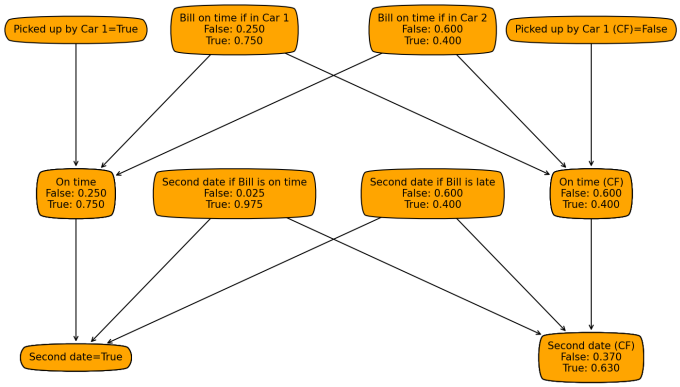

Figure 5: Bayesian network representation of the counterfactual example where Bill was picked up in Car 1 and arrived on time and we query on if Bill were picked up by Car 2 instead.

This result does not seem to quite match up with our intuition. If we reason through it, they managed to have a second date and Bill was picked up in Car 1. Since we do not know whether Bill was on time or not, there are two possible scenarios: Ann agreed to go on a second date even though he was late or Bill arrived on time and Ann agreed to go on a second date. So, we should increase the probability of Ann agreeing to go on a second date for both scenarios, where Bill is on time and where he is late. He is also more likely to be on time, to increase the chance of them going on a second date. So, if he is more likely to be on time, we should update the probability of him being on time in Car 1. However, as mentioned before, if we know that Car 1 and Car 2 are identical in every way and the environment is also completely identical, which includes traffic patterns, weather, etc., then it is intuitive for us to infer that him being more likely to be on time in Car 1 tells us he is also more likely to be in Car 2 as well. But, if we compare the probability of Bill being on time in Car 2 in Figure 4 with the probability in in Figure 1, Bill has the same probability of being on time in Car 2 when we know that he got a second

date after riding Car 1 as when we didn't know anything at all. Similarly, his probability of being on time in the counterfactual portion of the model is also identical to the model when we make no observations. This seems strange – our knowledge of Bill going on a second date when conditioned on being in Car 1 tells us *nothing* about the scenario where he is in Car 2, even though we intuitively believe that knowing this information in Car 1 *should* tell us something about being in Car 2, given that Car 1 and Car 2 are identical and the environment they are situated in are also identical.

This strange outcome arises from the way we have defined our independent noise variables. In this example, we have defined our noise variables as being independent from one another. That is, Bill being on time in Car 1 is independent from Bill being on time in Car 2. But as mentioned above, intuitively information about arriving on time in Car 1 tells us information about arriving on time in Car 2. We expect this because Car 1 and Car 2 are identical, but also because the environment that both cars are driving in are identical – so we expect to know more about the effect traffic might also have on Bill's arrival time.

This way of representing the independent noise variables is used in Balke and Pearl's paper in 2013 [1] and in applications of their twin model [5]. Their earlier work [2] used a different definition, however, where the child is a deterministic response function by mapping the parent's domain to the child's domain. The function then has a different output for each combination of the parent's value and noise. This representation avoids this counterintuitive result by not treating the noise as independent given the parent, but instead must exhaustively provide probabilities for each combination of parent and noise value.

Conclusion

Much of the literature surrounding Pearl's twin model theory of counterfactuals focus on applications of his theory, but overlook properties that may arise in the definition or structure of the network. Our goal was to use a simple example to demonstrate (1) the parents of the observed variable in the counterfactual scenario dose not need to be represented and (2) simply using the independent conditional probability for noise may lead to counterintuitive results if one overlooks the independence assumption built into the definition. The results highlight the importance of choosing a parameter for the model which best represents the world. As discussed in the addendum, this means one should also not blindly assume that the response-function noise variable definition is *the* best representation in all cases either. By being aware of the assumptions made in our models and parameters, we can also better understand the potential effects of using a particular representation and reasoning system – both desired and undesired.

Addendum

An important note about the outcome with the independent noise variables is that it highlights the difference between counterfactuals and intervention in Pearl's causal hierarchy. Numerically, if we were to observe Bill being picked up by Car 1, or even intervene, we still cannot capture the causal relationship between Bill being picked up by either car on him being on time because they are probabilistically independent but causally dependent. This particular relationship is the reason why the inferences made through the updates in the Bayesian network do not match our intuition, because we know information about the causal relationship of events beyond what we infer by measuring our probabilities and results numerically. After all, if we did not know more information about how Car 1 and Car 2 were identical in identical traffic, we would have no reason to believe that knowing one should affect our belief about the other.

It is important to also note that a functional response variable is not necessarily the best way to model noise. In Pearl's book [3] and paper[2], he states that specifying a functional model is arguably needed in the Bayesian network specification process. While a functional model is more descriptive and avoids the issue of not updating the noise variables appropriately in our date example, it has its own issues.

Firstly, a functional noise model requires far more specifications. For example, the noise function in Pearl's example in his 1994 paper [2] for Bob's probability of attending the party requires 4 probabilities to be specified, which has one binary parent. This probability function grows exponentially with the domain and parents, so it may not be feasible to fully represent this nor may it be worth the cost of attaining and updating with new evidence.

Secondly, there is not one singular best model for any given problem. When we represent counterfactual arguments/scenarios as Bayesian networks (or any other sort of representation), there are many ways to approach the problem with various trade-offs. As mentioned above, storage and computational cost may be a strong motivation to opt for the independent noise functions. But, if you know that your usage of Bayesian networks has a similar issue to the date scenario, it might be better to represent them in the response function way, or in another way. For example, an example that has the same causal structure as our date example is one where we can choose between two slot machines, both of which have the same rate of winning. If you win, you get more tickets, so you have a higher chance of winning the raffle afterwards. But if you lose, you get fewer tickets, so you have a lower chance of winning the raffle. The Bayesian network would look like the one below.

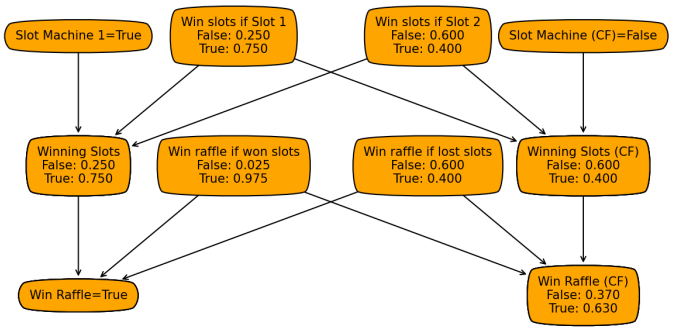

Figure 6: Bayesian network representation of a counterfactual where you play slots on the first machine and win the raffle, querying on if you had played with the second machine.

Intuitively, this feels different from the date example. Being more likely to win slots with the first machine does not actually tell us anything about our likelihood of winning slots with the second machine. Since they are completely independent in their mechanisms of obtaining their winning result (whether it be a random number generator, physical machinery, etc.), we do not expect our belief about the first machine's chance of winning to affect our belief about the seconds machine's chance of winning.

So, treating the noise variables as independent from each other in this scenario makes sense because the probabilities seem to be independent based on what we know about the world. Meanwhile, the response function definition of the noise variables for the date example seems to suit the problem better since we want to avoid treating them as independent.

References

[1] Alexander Balke and Judea Pearl. *Counterfactual Probabilities: Computational Methods, Bounds and Applications*. 2013. arXiv: 1302.6784 [cs.AI].

[2] Alexander Balke and Judea Pearl. "Probabilistic evaluation of counterfactual queries". In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 1994, pp. 237–254.

[3] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[4] David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2023.

[5] Athanasios Vlontzos, Bernhard Kainz, and Ciaran M. Gilligan-Lee. *Estimating Categorical Counterfactuals via Deep Twin Networks*. 2023. arXiv: 2109.01904 [cs.LG].