Diving for Causal Pearls

Johannes M. Halkenhaeusser

Minerva Schools at KGI

IL181.006 – Prof. Scheffler

Fall 2020

If you have spent enough time around researchers in social sciences, you have, at one point, heard about DAGs. Most likely, you ignored it as just another random acronym relevant to only a subfield of your area of research. However, DAGs or Directed Acyclic Graphs are a powerful way to visualize causal relationships, translate them to/from real structural equation models, use them to derive methods of causal inference, or cast them to linear regressions. But let's take it one step at a time to go through those features and look at how DAGs can help us think about interventions and calculate the causal effect of coffee on our productivity using inverse probability weighting.

Coffee three ways

Researchers like to talk about their models, assumptions, and hypothesis in roundabout or jargon-infested ways that can be confusing. But DAGs clear up the confusion. If we wanted to test hypothesis that coffee (X) increases the number of words written for an assignment (Y). We suspect that X and Y are affected by the number of sleeping hours we had the day before (Z). Coffee also changes our level of happiness (H). First, we can represent our hypotheses in a DAG in which each variable is denoted by a node and the direction of effect is denoted by the arrows (Fig. 1).

A second way we can express the relationship by decomposing it into a structural equation model (SEM) consisting of exogenous and endogenous variables that are connected through functions. Exogenous variables are those that are not defined by any other variable in the graph, meaning they have no arrows pointing toward them (technically they are further defined by another exogenous random distribution but we will leave this out for simplicity). Endogenous variables are defined by the exogenous variables through functions F. Fig. 1 can be decomposed into an SEM. Notice how suddenly our DAG lead to some neat linear equations popping out in the functions. They almost look like something we could use in a regression.

$Exogenous$: {Z}

$Endogenous$: {X,Y,H}

Functions:

- $f_X : M = \alpha Z$
- $f_Y : Y = \omega X + \gamma Z$
- $f_H : H = \theta X$

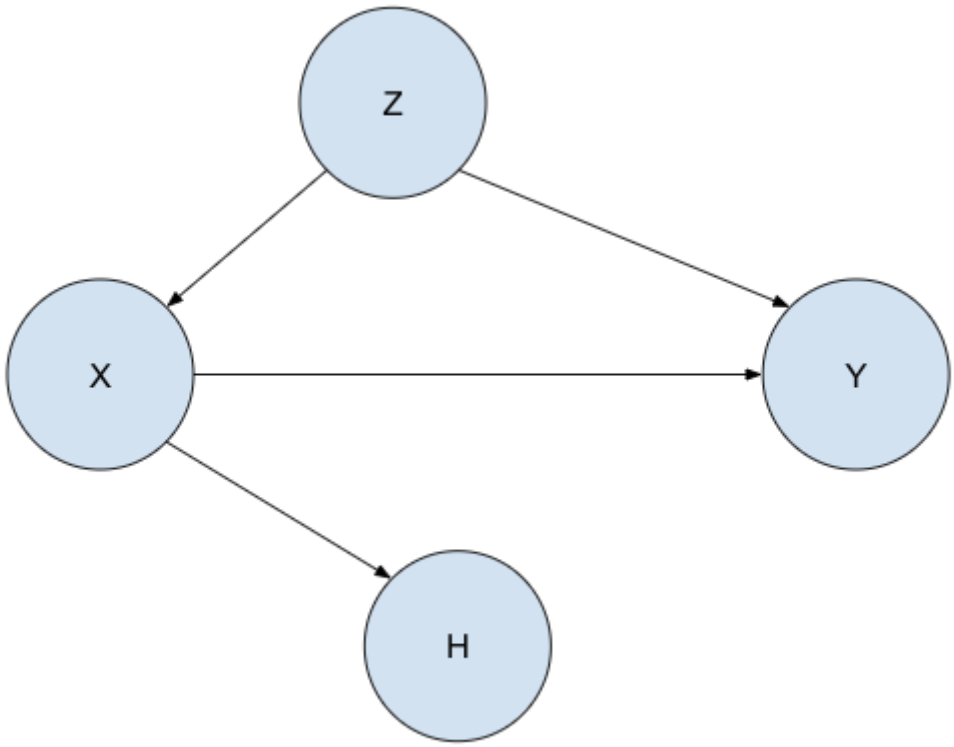

Figure 1. A basic DAG where our exposure of interest X affects Y and H. Z affects both X and Y.

Thirdly, through product decomposition we can express the relationships in the model in terms of one another's probabilities. That means you can read the joint probability of the variables off the graph by using the product rule:

$$P(x_1, x_2, \dots, x_n) = \sum_i P(x_i|pa_i)$$

where pa_i are the parents (all the nodes pointing into x_i) of the variable. It is easiest to go from exogenous to exogenous variables. Hence we start with laying a probability distribution over the hours of sleep P(Z) and multiplying that with the distribution over our coffee intake given the amount of sleep we got P(X | Z). As we make our way through the model we end up with:

$$P(X, Y, Z, H) = P(Z)P(X \mid Z)P(Y \mid Z, X)P(H|X)$$

Decomposing the mode lets us cast our caffeine-induced theory in light of probabilities that lets us express uncertainty within the model and depict that in the real world, hours of sleep are not an integer value but normally distrubted around some average. Notice how our research question necessitates making a modeling choice that the coffee affects words written and how the DAG clearly visualises this choice. With the example of coffee, this may seem inconsequential but consider a recent paper testing if being part of a minority leads to an increased experience of

police force (Knox, Lowe, & Mummolo, 2020). In politically charged subject areas like this, creating a DAG that clearly expresses assumptions and the complex interplay of variables allows for clarity. It also enables us to detect variables that may be confounding our causal relationship or have brought about selection bias.

Coffee, Colliders, and Confounders

Selection bias occurs when we control for a variable that is the common effect of two variables that are independent. In statistics lingo, they become conditionally dependent as we condition on a collider.

$$P(X|collider) \not\perp\!\!\!\perp P(Y|collider)$$

Imagine that we are happy (H) whenever it is sunny (S) or we have had coffee and that the number of words written also somehow depend on the amount of sun. However, we consume a sun-independent amount of coffee (Fig. 2). If we condition on the level of happiness, we introduce a spurious correlation between coffee and the level of sunshine. Usually, sun would not tell us anything about our coffee consumption or words written, but because we need coffee or sun to be happy and have information about our happiness, we can make inferences about X by knowing S. They have become conditionally dependent through a so-called collider, H. This means that there is a new path between X and Y, that goes through H and S.

A relationship between two variables is confounded if the variables have a common cause. In our example, the amount of sleep (Z) we got affects both our treatment (coffee) and outcome (words). In a world where we do not account for the amount of sleep we have gotten, and no effect of X on Y, we may attribute the effect sleep has on the words written to coffee. Consequently, we would grossly misjudge the effect and see our hypothesis as proven through a spurious correlation that is by no means causal.

To deal with confounders, we have to control for confounders or variables between them and the treatment or outcome variable. We should avoid conditioning on colliders and their descendants or block the path by conditioning on a variable within the path (controlling for descendants of colliders also renders the income variables conditionally dependent). Once we have accounted for these sources of spurious correlations, we have satisfied the backdoor criteria:

" **Backdoor Criterion**:

Given an ordered pair of varaibles (X, Y) in a directed acyclic graph G, a set of varaibles Z satisfies the backdoor criterion relative to (X, Y) if no node Z is a descendant of X, and Z blocks every path X and Y taht contains and arrow into X."

Pearl, J., Glymour, M., & Jewell, N. P. (2016)

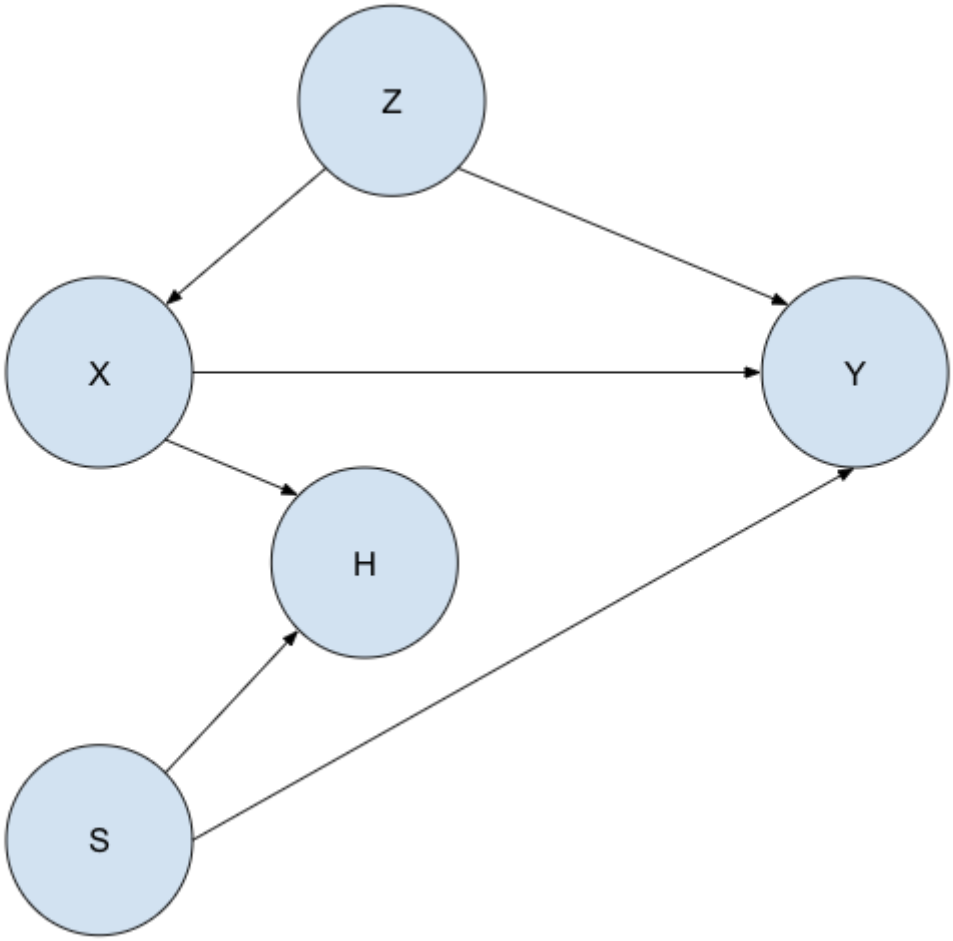

Figure 2. The collider H when conditioned creates a confounded correlation between X and Y. Hence, we should avoid conditioning on H or if we have to, condition on S also.

Causal Effect Rule

If we now wanted to estimate the causal relationship between coffee and words we can use the causal effect rule:

Given a graph G in which a set of variables PA are desginated as the parents of X, the causal effect of X on Y is given by:

$$P(Y|do(X = x)) = \sum_z P(Y = y|X = x, PA = z)P(PA = z)$$

where z ranges over all the combinations of values that the variables in PA can take.

Notice the do(X = x) operator in this equation, which signals that we are carrying out an intervention. An intervention occurs when the variable X is forced to take on some value. This fundamentally changes the graph as now it is not the parents of X that determine its value but the intervention. Graphically, we delete all the edges going into X (Fig. 3). Through summing over all the combinations of z, we adjust for them mathematically. We can estimate the causal effect

of treatment by calculating the difference between the outcome probability with two different interventions. This is known as the Adjustment Formula:

$$P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0))$$

The causal effect rule can also be rewritten as:

$$P(y|do(x)) = \sum_z \frac{P(X=x,Y=y,PA=z)}{P(X=x|PA=z)}$$

The denominator P(X=x| PA=z) is the probability of being treated given the a set of other variables. This is also called the propensity score which we can use in a causal inference method

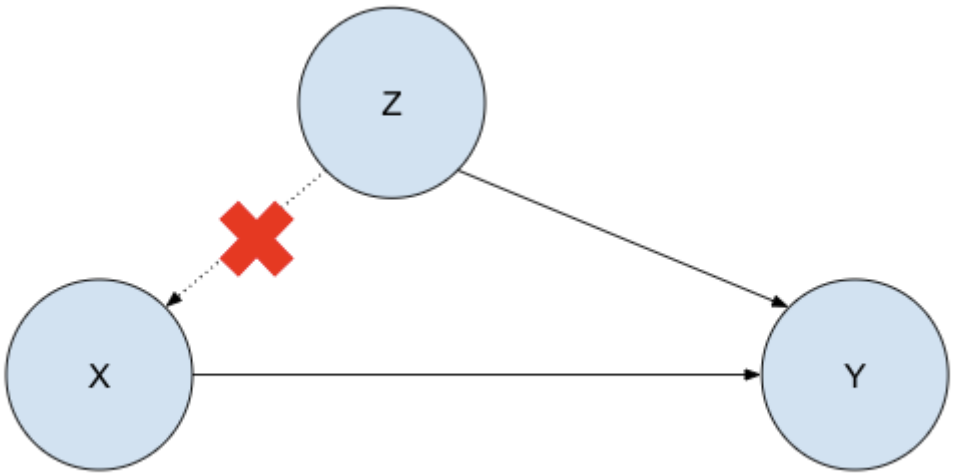

Figure 3. Our original model with an X intervened on. Note that S and H have been left out as we are not conditioning on the collider and S hence becomes an irrelevant exogenous variant. As the backdoor path is closed we can concentrate on accounting for the confounder Z

Inverse probability weighting

When we use the regular causal effect rule, we have to sum over each combination of the parent variables of X. Now consider a system that has many parents with many different variables that take many discrete values. The amount of data for each stratum and computational capacity needed skyrockets. Instead we can estimate the propensity score through, for example, logistic or linear regression to determine the likelihood of being treated for each observation. The goal is to balance the treatment probability between the treatment and control groups as to make the effect of the parent influences irrelevant. Inverse probability weighting reweights every observation with its propensity score and adds them up to create a weighted average. Let's examine it in a toy example of our coffee drinking scenario.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import scipy.stats as sts
5 import random as rd
```

```
from sklearn import linear_model
```

```
n = 10000

#first we randomly choose an exogenous amount of sleep
sleep = sts.randint.rvs(7,9,size =n)

#the probability of drinking coffee is a function of sleep
coffee_given_sleep = [0.67 if i == 7 else .33 for i in sleep]

#see who is drinking coffee
coffee = np.asarray([1 if sts.uniform.rvs() < i else 0 for i in coffee_given_sle

#get the probability of writing any words given coffee drinking and sleeping
#Coffee increases the probability of words written by 0.25 and an extra hour of
words_given_sleep_coffee = 0.25 * coffee + 0.05*(sleep - 7) + .05*sts.norm.rvs(s

#see who is drinking coffee
words = np.asarray([1 if sts.uniform.rvs() < i else 0 for i in words_given_sleep
```

As shown in the data generating process, sleeping 7 vs 8 hours increases the probability of drinking coffee by 0.34 but less sleep also decreases the probability of sleeping. Coffee increases the probability of writing words by 0.25. This 0.25 is the direct effect of coffee we are interested in. The joint probabilities of each combination of the variables is shown in the table below. Let's see if we can estimate the effect of coffee by reweighting our observations with their appropriate propensity score.

```
table = pd.DataFrame({'Sleep':sleep, "Coffee":coffee, "Words" :words, "Real Prop
joint_p = table.groupby(['Coffee', "Words", "Sleep"]).size().reset_index().renam
joint_p.Share /= n
joint_p.rename(columns={'Share' :"P(X, Y, Z)"}, inplace = True)
joint_p
```

	Coffee	Words	Sleep	P(X, Y, Z)
0	0	0	7	0.1602
1	0	0	8	0.3166
2	0	1	7	0.0037
3	0	1	8	0.0202
4	1	0	7	0.2502
5	1	0	8	0.1135
6	1	1	7	0.0839
7	1	1	8	0.0517

Table 1: The different combinations of sleeping hours, coffee, and words written and their joint

```
#Estimate the propensity score using a linear regression model
p_model = linear_model.LinearRegression()
p_model.fit(sleep.reshape(-1, 1), coffee.reshape(-1, 1))
print("Our estimated linear coefficient is:", p_model.coef_[0,0])

```

```
Our estimated linear coefficient is: -0.34179986879790336
```

This is pretty accurate. Next we estimate the propensity score for the different combinations of our variables. One could make a weighted average of all the individual observations but we apply the inverse probability to the joint probability as it is faster in this case than reweighing every observation individually. We assume/know that sleep and all other exogenous variables are evenly distributed among the groups.

```

weighted = []

#for each combination of sleepers we weight the different joint probabilities wi
for i, row in joint_p.iterrows():

    #calculate the propensity score
    p_score = p_model.predict(np.asarray(row['Sleep']).reshape(1, -1))

    #weight the joint probability
    weighted.append(row['P(X, Y, Z)']/p_score[0,0])

#let's add it to our table
joint_p['Weighted'] = weighted
```

```
joint_p
```

	Coffee	Words	Sleep	P(X, Y, Z)	Weighted
0	0	0	7	0.1602	0.238790
1	0	0	8	0.3166	0.962065
2	0	1	7	0.0037	0.005515
3	0	1	8	0.0202	0.061383
4	1	0	7	0.2502	0.372941
5	1	0	8	0.1135	0.344897
6	1	1	7	0.0839	0.125059
7	1	1	8	0.0517	0.157103

Table 2. The joint probabilities and the weighted joint probabilities

```
final_table = pd.pivot_table(joint_p, ['Weighted'], ["Coffee", "Words"],aggfunc
final_table
```

		Weighted
Coffee	**Words**	
0	**0**	1.200855
	1	0.066898
1	**0**	0.717838
	1	0.282162

Table 3. The aggregated weighted joint probabilities. We only care about the probabilities where Words = 1 for our causal effect.

Now using the Adjustment formula

$P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0))$:

```
print("Causal Effect of Coffee ~", final_table["Weighted"].values[3] - final_tab
```

```
Causal Effect of Coffee ~ 0.21565125974805055
```

There is still some error in the estimation which creeps in from the non-deterministic portion of the model. Inverse probability can lead to high weights for those observations very low propensity score such as 0.001. This means that some few observations have a high weight in the final calculations which can introduce error if there are other confounders or unobserved treatment predictors.

To summarize, we have seen how we can use a DAG to depict our knowledge and assumptions about causal relationships between variables and how to cast a DAG to structural equation models. After understanding the problem of confounders and colliders, we moved on to a brief introduction to interventions. This allowed us to estimate the causal effect through inverse probability weighting.

Works Cited

Gonçalves, B. (2020, 07 25). Structural Causal Models. Retrieved from Medium: https://medium.com/data-for-science/causal-inference-part-iv-structural-causal-models-df10a83be580

Knox, D., Lowe, W., & Mummolo, J. (2020). Administrative records mask racially biased policing. American Political Science Review, 114(3), 619-637.

Pearl, J., Glymour, M., & Jewell, N. P. (2016). Causal inference in statistics: A primer. John Wiley & Sons.

HCs

- #correlation: The whole article explains how to distangle correlations. By explaining how the dependencies between variables can cause confounding, collider bias, or be used to figure out the causal effect I made gave examples of how to distinguish between correlation and causation.
- #organization: I had to use organization to cover the thematic points of this article. I wanted to give a solid foundation of what DAGs are and build up the complexity from very easy (understanding what a DAG is), to harder (giving more formulas), on to applied coding. This allowed the article to flow and naturally come to an ending that went away from the abstract SEM-lingo towards a easy to understand example.

```
1
```