PhD notes Strategy of Causal Interventions

Maarten Vonk

September 10, 2021

1 Introduction

This notebook is meant to serve as structured notes for my effort to pursue a PhD in causal inference. It is not a formal introduction to causal inference, but more an intuitive approach to concepts central in causal inference. It also serves to contextualize the literature within the broader field of causality. Articles are linked together to provide insight in how one research builds upon earlier results. Examples and simplifications of the concepts introduced in literature are provided to better digest these concepts. Some concepts have been worked out completely while other research has just been contextualized within broader literature.

For applications of some of these concepts, some software (packages in Python/R) already exists. In order to get more familiarized with these concepts and the existing application packages, links to Colab Notebooks are included in the text with applications or further analysis. This notebook is a work in progress, meaning that errors can be present and the structure is not always optimized.

There is not a chapter with all the preliminaries, because many articles use a specific set of preliminaries that can best be included in the corresponding chapter. However some essential preliminaries of causal inference are included in chapter 2.

Contents

2 Introduction to Causal Graphs

This chapter is meant to offer some preliminaries in probabilistical graphical models. These preliminaries are necessary components for causal inference, because they give rise to the Bayesian networks, which form the basics of causal models. Also, this will help provide intuition to understanding the rules of do-calculus. The assumptions that combine probability theory and graph theory are stated explicitly. Many have been extracted from the Causal Textbook of Brady Neal [3], because it offers an accessible introduction to causal inference.

2.1 Graph Terminology

A graph is denoted by $G = (V, E)$ where V is the set of vertices and E the set of edges. A graph can be *directed* when every edge has a direction, *undirected* when no edge has a direction or *partially directed* when some but not all edges have a direction. A graph can contain a *cycle* when there exists a directed path from a node to itself. When there is no such path and the graph is directed, we call this a *directed acyclic graph* or DAG.

When $X \to Y$, we say that X is a parent of Y and Y a child of X. The set of parents of Y is denoted by $\text{pa}(Y)$ and the set of children of X is denoted by $\text{ch}(X)$. An ancestor of Y is a node with a directed path to Y, including Y itself. The set of ancestors is denoted by $\text{an}(Y)$. Similarly a descendent of Y is a node with a directed path from Y including Y itself and the set is denoted by $\text{de}(Y)$. Finally we distinguish a couple of different directed graph structures: We call the structure $X \to Z \to Y$ a *chain*, $X \leftarrow Z \to Y$ a *fork* and $X \to Z \leftarrow Y$ a *v-structure*. In the last case Z is called the *collider*.

2.2 Bayesian Network

We now extend this graph knowledge to introduce probabilistic models. According to chain rule of probability, we can write the distribution $P(x_1, \dots x_n)$ in terms of its factors:

$$P(x_1, \dots x_n) = P(x_1) \prod_i P(x_i \mid x_{i-1} \dots x_1)$$

but we can simplify this if we assume the following:

Assumption 1 ***Local Markov Assumption.*** *Given a DAG G, a node X is conditionally independent from all variables except its parents $pa(x_i)$ and children.*

This assumption allows us to write the joint probability in a much more tractable way:

$$P(x_1, \dots x_n) = P(x_1) \prod_i P(x_i \mid \text{pa}(x_i)).$$

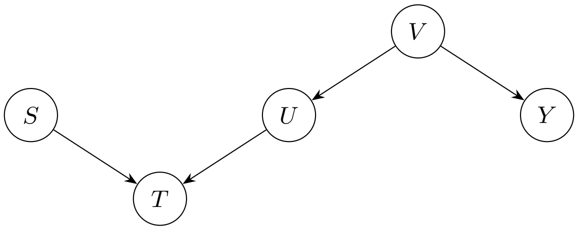


Figure 1: DAG

This is called the *Bayesian Network Factorization* and we call P and G *Markov compatible* or distribution P is *Markov* to G. Now in order to completely connect graphs to the probability distribution, we need one more assumption that incorporates dependencies:

Assumption 2 ***Adjacency-Faithfulness Assumption.*** *Given a DAG G and suppose that two variables are adjacent in the G. Then they are not independent conditional on any subset of other variables.*

Both assumptions can be combined to obtain the minimality assumption:

Assumption 3 ***Minimality Assumption.*** *Given a DAG G and suppose that distribution P is Markov to G. Then no proper subgraph of G is also Markov to P.*

In other words, the remaining edges imply dependency between variables. Because we are dealing with directed edges, these dependencies do not have to be bilateral but can be one-sided:

Assumption 4 ***Causal Edges Assumption.*** *Given a DAG G, every directed edge from X to Y in G implies that X has a causal effect on Y.*

2.3 Dependencies in Graph Structures

We create distributions according to a graph that contains a chain, fork and v-structure. In each of these cases we will derive the dependencies and the conditional dependencies of the variables. We include the python code to illustrate the true distribution in this Colab Notebook, but first we share the graph that lie underneath the distributions in Figure 1.

Based on this graph we create distribution where T depends on S and U, U depends on V and Y depends on V:

Listing 1: Distributions

```
N = 5000
s = np.random.uniform(size=N)
v = np.random.normal(size=N)
u = 2. * v + 0.1 * np.random.normal(size=N)
t = 2. * s + u + 0.1 * np.random.normal(size=N)
```

```
y = np.random.binomial(1., p=1./(1. + np.exp(-5. * v)))
df = pd.DataFrame({'S': s, 'T': t, 'U': u, 'V': v, 'Y': y})
```

We can check dependence by simple pearson correlation test to obtain the following results shown correlation plot in Figure 2. As we expect according to our distributions: S and T are dependent, but S is independent from all other variables since the collider T blocks all association. The variable Y is dependent on all other variables except S since association goes through the fork structure $U \leftarrow V \rightarrow Y$.

Note that all that we have checked so far is unconditional dependence. We are also interested in dependencies when we condition on variables. The various structures; chain, fork and v-structure behave different under conditional dependence. First the dependence under the chain structure $V \rightarrow U \rightarrow T$ when we do not condition (we know we have dependency, but we check again to show the difference in results) and then we check again when we condition on the mediator node U. We do the same exercise for the fork and the v-structure to obtain the following:

Listing 2: p-values Chain Dependence (rounded to 6 decimals)

```
V and T are unconditionally dependent cause p-value<0,05:   0.0
V and T conditional U are independent cause p-value>0,05:   0.664282
```

Listing 3: p-values Fork Dependence (rounded to 6 decimals)

```
U and Y are unconditionally dependent cause p-value<0,05:   0.0
U and Y conditional V are independent cause p-value>0,05:   0.42431
```

Listing 4: p-values v-structure Dependence (rounded to 6 decimals)

```
S and U are unconditionally independent cause p-value>0,05: 0.312844
S and U conditional T are dependent cause p-value<0,05:       1.5e-05
```

The conclusion of this exercise is the following: in chain and forks we have unconditional dependence between the distant variables, but independence conditional on the middle variable. In v-structure we have unconditional independence between the distant variables, but conditional dependence on the middle,

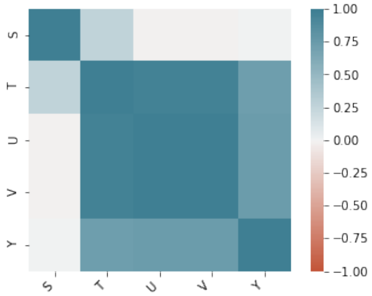


Figure 2: Correlation Plot

collider variable (or any descendent of the collider variable). This concept gives rise to the definition of *d-separation*.

Definition 1 ***d-separation*** *A path p is blocked by a set of nodes Z if and only if*

1. *p contains a a chain $T \to U \to V$ or fork $T \leftarrow U \to V$ where U is contained in Z.*

2. *p contains a a v-structure $T \to U \leftarrow V$ and the collider node U or any of the descendants of the collider is in Z.*

If Z blocks every path p between two nodes X and Y, then X and Y are d-separated by Z: $X \perp\!\!\!\perp Y \mid Z$.

Given the graph in Figure 1, S and U are d-separated given the empty set, but S and U are not d-separated given T. The formally state the connection between probability independence and graph independence we have the following Theorem.

Theorem 1 ***Global Markov Assumption*** *Given that P is Markov with respect to G, if X and Y are d-separated in G conditioned on Z, Then X and Y are independent in distribution P conditioned on Z.*

$$X \perp\!\!\!\perp_G Y \mid Z \Rightarrow X \perp\!\!\!\perp_P Y \mid Z.$$

Even though this is framed as a theorem, this can also be framed as an assumption since it is derived from the local Markov assumption.

3 Introduction to do-operator

In order to say something about causal effects, we should first say something about interventions. Therefore we should introduce the do-operator and do-calculus that goes along with that. This has been developed by Judea Pearl [1]. When we condition on a certain variable, say $T = t$, we are saying we are restricting ourselves to look at the subset of observations where $T = t$. However when we say we intervene on variable $T = t$, we say we enforce t to the entire population. See Figure 3 from Brady Neil's textbook for a clear distinction.

The intervention is specified by the do-operator and is written as $do(T = t)$. We call the distribution following from the intervention the *intervention distribution:* $P(Y \mid do(T = t))$ or simply $P(Y \mid do(t))$. Similarly we call a distribution *observable* when there is no do-operator in the distribution. The relation between these two observable is described in the following definition:

Definition 2 *Suppose Q is an expression containing a do-operator. Then Q is identifiable if we can replace the expression containing the do-operator with an expression without a do-operator.*

We will get back at how to rewrite the expression containing a do-operator in an expression without do-operator when we introduce the do-calculus. We first have to introduce an additional assumption describing the locality of the intervention:

Assumption 5 *Modularity Assumption. Suppose we intervene on a subset of nodes S, then for all nodes i we have*

1. *If $i \notin S$, then $P(x_i \mid pa(x_i))$ remains unchanged*
2. *If $i \in S$, then $P(x_i \mid pa(x_i)) = 1$ with x_i being the value set by the intervention and 0 otherwise.*

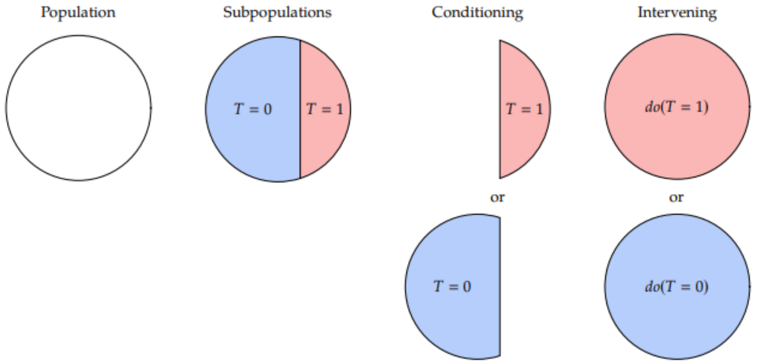


Figure 3: Difference Intervention and Conditioning

Note that the modularity assumptions encodes that the intervention is local in the sense that only the conditional distribution of X_i given its causes (parents) is affected by the intervention on X_i. Note that the modularity assumptions and bayesian network factorization allows us to rewrite an intervention distribution in the following way:

Theorem 2 *Truncation Formula: Assume that P is Markov to G and satisfies modularity. Given a set of intervention nodes S and x being consistent with S, we can write*

$$P(x_1, \ldots, x_n \mid do(S = s)) = \prod_i P(x_i \mid pa_i)$$

and of course $P(x_1, \ldots, x_n \mid do(S = s)) = 0$ if x is not consistent.

Using this new way of writing interventional distributions, we can bolster the intuition behind Figure 3 by formalizing the difference between intervening and conditioning on a variable. For this consider the causal graph in Figure 4. Assume that P is Markov to this graph, then according to the bayesian network assumption we can write the joint distribution.

$$P(y, t, x) = P(x)P(t \mid x)P(y \mid t, x)$$

We we start intervening, we can use Theorem 2 to rewrite the distribution as follows:

$$P(y, x \mid do(t)) = P(x)P(y \mid t, x)$$

and by marginalizing x out we receive

$$P(y \mid do(t)) = \sum_x P(x)P(y \mid t, x)$$

While the conditional probability $P(y \mid t)$ can be reframed as

$$P(y \mid t) = \sum_x P(y, x \mid t) = \sum_x P(x \mid t)P(y \mid t, x).$$

Hence the difference between interventional distribution and conditional distribution is the difference between $P(x)$ and $P(x \mid t)$ as is illustrated in Figure 3. Theorem 2 is known as the *truncation formula* and is very helpful in rewriting interventional distributions into statistical distributions, but the drawback is that we assume that we can calculate every conditional distribution. However, sometimes we are dealing with unobserved variables and we need the do-calculus to help identifying interventional distributions.

3.1 Intuition do-calculus

Now we can combine the concept of d-separation together with Markov assumption to arrive at the rules of do-calculus intuitively. For a proof of the rules see

[1] Consider a graph depicted in Figure 5. When we condition on Z, this means that X and Y become d-separated. As we have seen in the previous chapter, this implies that X and Y become independent in distribution when P is Markov to graph G. This means:

$$P(y \mid x, w) = P(y \mid w) \text{ if } Y \perp\!\!\!\perp X \mid Z.$$

This can then be generalized to form the first rule of do-calculus.

$$\textbf{Rule 1: } P(y \mid do(x), z, w) = P(y \mid do(x), w) \text{ if } (Y \perp\!\!\!\perp Z \mid X, W)_{G_{\overline{X}}}.$$

Now we are going back to Figure 4. Note that when we condition on X, the only relation there is between T and Y is a direct path or in other words: T become d-separated from Y when we remove all the arrows going from T, the last directed path. In this way intervening is the same as conditioning hence:

$$P(y \mid do(t), x) = P(y \mid t, x) \text{ if } (Y \perp\!\!\!\perp T \mid X)_{G_{\underline{T}}}.$$

which can again be generalized to the second rule

$$\textbf{Rule 2: } P(y|do(z), do(t), x) = P(y|do(z), t, x) \text{ if } (Y \perp\!\!\!\perp T|X, Z)_{G_{\overline{X}\underline{T}}}.$$

Rule 3 can also be intuitively derived, but can be a little harder. Suppose the graph of Figure 6. In this graph T and Y are d-separated when we intervene on T because this would break the arrow from V to T. This d-separation would suggest that intervening on T does not have affect on Y. Also, in this example

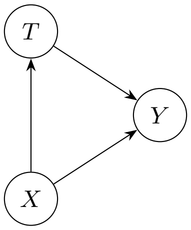


Figure 4: Simple Causal Graph

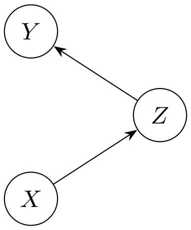


Figure 5: Graph

conditioning on W does still keep this d-separation intact, which would suggest the following:

$$P(y \mid do(t), w) = P(y \mid w) \text{ if } (Y \perp\!\!\!\perp T \mid W)_{G_{\overline{T}}}.$$

However, this is **NOT** true! This can be understood by taking a look at the modification of the graph in Figure 7. In this graph conditioning on W would imply that U and Y become dependent variables because W is the descendent of a collider. Consequently, intervening on T would break that dependence and therefore we cannot just remove the do-operator in this equation. Instead we can check if we can remove the do-operator by checking those nodes in T that are not ancestors of the W we are conditioning on. In this way, we make sure that conditioning on W does not create additional dependencies that we consequently break with the intervention. Hence the correct way of formulating this is:

$$P(y \mid do(t), w) = P(y \mid w) \text{ if } (Y \perp\!\!\!\perp T \mid W)_{G_{\overline{T(W)}}}.$$

where T(W) are the nodes of T that are not ancestors of W. Again, generalizing this yields the last rule:

Rule 3: $P(y \mid do(z), do(t), w) = P(y \mid do(z), w)$ if $(Y \perp\!\!\!\perp T \mid W, Z)_{G_{\overline{T(W)Z}}}$.

The rules of do-calculus shed a new light on Definition 2, meaning the rules are the toolkit needed to rewrite an expression containing a do-operator to an expression not containing a do-operator. In fact, Pearl [2] proved that the rules of do-calculus are complete. This means that there is no identifiable expression containing a do-operator who cannot be rewritten to an expression without a do-operator using the three rules of do-calculus. We return to identification and factorization of do-calculus in a later chapter. First we are going to apply the do-calculus to see how this works in practice.

3.2 Soft and Hard Interventions and σ-calculus

So far we have considered one sort of intervention. In contemporary articles, researchers start to make a distinction between soft (parametric) and hard (structural) interventions. Because this will come back in later chapters, we consider

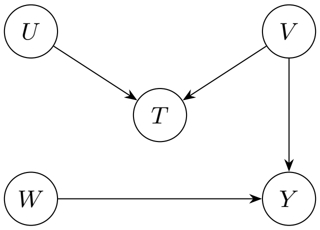


Figure 6: Causal Graph 1 for intuition rule 3

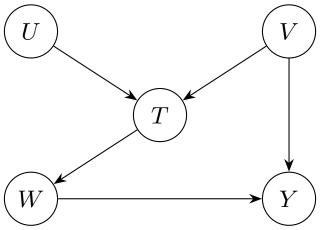


Figure 7: Causal Graph 2 for intuition rule 3

the difference here by means of the article of Frederick Eberhardt and Richard Scheines [31] and [32]. While [31] are one of the first to formally state the separation between the different sorts of interventions, [32] generalized the concept of do-calculus into σ-calculus that allows for identification of causal effects when considering soft interventions.

Soft and Hard interventions Up to this point we have considered only hard or atomic intervention. In this type of intervention, intervening means that we set a variable to a specific value and hence we break the relation between the intervened variable and the variables that have a direct causal effect on the intervened variable. As [32] correctly points out, this is often a unrealistic scenario. It assumes that if we imply an intervention, we assume that the policy underlying the intervention has a flawless effect. For example we can derive a policy that makes a complete society smoke-free. Much more realistic is if the policy has a much more nuanced effect, for example the policy intervention can reduce smoking with 40 percent. Such an intervention leaves the underlying causal mechanism intact, but the distribution of the variable conditional on its causes is changed. Such an intervention is called a *soft intervention.* We can state the difference more formally with [31]:

A hard intervention I on a variable X in a system of variables V is an intervention for which:

- I can only be on or off.
- When I is off, the observational distribution over V holds.
- The intervention I intervenes on X only.
- The intervention I itself has no causes.
- When the intervention I is on, X will be independent of all its causes in V and in the factorized distribution $P(V)$, the term $P(X \mid \mathrm{pa}(X))$ will be replaced by $P(X \mid I)$.

For hard interventions, the do-calculus as described above applies.

A soft intervention I on a variable X in a system of variables V is an intervention for which:

- I can only be on or off.
- When I is off, the observational distribution over V holds.
- The intervention I intervenes on X only.
- The intervention I itself has no causes.
- When the intervention I is on, X will *not* be independent of all its causes in V and in the factorized distribution $P(V)$, the term $P(X \mid \mathrm{pa}(X))$ will be replaced by $P(X \mid \mathrm{pa}(X), I)$.

In contrast to hard interventions, the do-calculus does not apply to soft interventions in the same way. This is the reason that [32] generalized the do-calculus to σ-calculus in a way that still allows for identification of causal effects.

The difference between the hard and soft intervention amounts to the difference between eliminating the arc from Z to X in the intervened graph depicted in Figure 8. Note that I is denoted by dashed lines since it is considered exogenous.

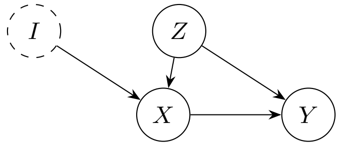


Figure 8: Example Intervention in Causal Graph

4 Application of the do-calculus and estimation

Now consider the example of Figure 4 again. In this chapter we are going to use the do-calculus to identify the causal estimate and we use rules from do-calculus to create two different estimates: the Inverse probability weight (IPW) estimator and the conditional outcome model (COM).

We define the average causal treatment effect ATE for binary treatment T on outcome Y as:

$$\mathbb{E}(Y \mid do(T=1)) - \mathbb{E}(Y \mid do(T=0)).$$

Suppose now we have the following observations: we have N datapoints with covariates X where X is uniformly distributed between $[0,1]$: $x \sim U(0,1)$. Treatment T has binomial distribution $t \sim B(1, \frac{1}{1+e^{-5x}})$. Also suppose we have outcome based on covariate X and treatment T: $y \sim 2x + t + 0.1\mathcal{N}(0,1)$.

4.1 Application do-calculus for identifiability

We can now prove that in our causal graph, Y is identifiable by treatment T. First we marginalize out x by rules of probability:

$$P(Y|do(T=t)) = \sum_{z} P(y|x, do(t))P(x|do(t)) \tag{1}$$

Now we we observe that $((Y \perp\!\!\!\perp T)|X)_{G_{\underline{T}}}$ and we apply the second rule of do-calculus to rewrite the first term:

$$P(y|x, do(t)) = P(y|x,t). \tag{2}$$

Similarly, we observe that $(X \perp\!\!\!\perp T)_{G_{\overline{T}}}$ to apply the third rule of do-calculus to obtain

$$P(x|do(t)) = P(x). \tag{3}$$

We can rewrite our first expression than to:

$$P(Y|do(T=t)) = \sum_{x} P(y|x,t)P(x). \tag{4}$$

Since the last expression is 'do-operator free', this is now an application of the do-calculus to prove identifiability. In this case the identifiablility is called the *back-door theorem.* There is also a *front-door theorem.* The derivation for this can be found in this colab notebook.

4.2 Application IPW algorithm for estimation

In this subsection we use the IPW algorithm to solve calculate the Average treatment effect (ATE). The python code is in this Colab Notebook. Note that we want to know

$$P(y|do(T=t)) = \sum_z P(y|t,z)P(z) = \sum_z \frac{P(y|t,z)P(t|z)P(z)}{P(t|z)} = \sum_z \frac{P(y,t,z)}{P(t|z)}$$

where the first identity follows from the do-calculus as shown above. This means we can obtain the intervention distribution $P(y|do(T=t))$ by accounting for the propensity score $P(t|z)$. So now we need to train a model $\hat{e}$ to predict T from Z. This is done by logistic regresion. We apply this models to obtain weights for each data point: $w_i = \frac{1}{P(T_i=t|z_i)}$.

Using this weights we will now sample from the intervention distribution $P(y|do(T=t))$ by sampling from our data given the weights we created to adjust for the covariate Z. Data points with higher weights w_i are more likely to get picked from the data than datapoints with lower weights. Note that there will be duplicates in our new data when we create datasamples of equal length to the original data. Since we had more observations of $T=1$ in our original dataset, observations of $T=0$ received higher weights and therefore the new dataset will be balanced.

Now given the causal relation is not confounded anymore, we can calculate the causal effect. Note for the expectation first that

$$\begin{aligned}\mathbb{E}(Y \mid do(T)) &= \sum_y \sum_z \frac{YP(y,t,z)}{P(t|z)} = \sum_y \sum_z \frac{yP(y)\mathbb{1}_{T=t,Z=z}}{P(t|z)} \\ &= \mathbb{E}(\sum_z \frac{Y\mathbb{1}_{T=t,Z=z}}{P(t|z)}) = \mathbb{E}(\frac{Y\mathbb{1}_{T=t}}{P(t|Z)}).\end{aligned}$$

So we we implement that to calculate the causal effect we have

$$\mathbb{E}(Y \mid do(T=1)) - \mathbb{E}(Y \mid do(T=0)) = \mathbb{E}(\frac{Y\mathbb{1}_{T=1}}{P(t|Z)}) - \mathbb{E}(\frac{Y\mathbb{1}_{T=0}}{1-P(t|Z)})$$

We can then estimate the causal effect by the following estimand:

$$\hat{\tau} = \frac{1}{n}\sum_i (\frac{y_i\mathbb{1}_{T=1}}{\hat{e}(z_i)} - \frac{y_i\mathbb{1}_{T=0}}{1-\hat{e}(z_i)})$$

where $\hat{e}$ is the model used to estimate $P(t|Z)$.

4.3 Application COM estimiation

There is another way of calculating the average causal treatment effect that also makes use of the back-door theorem, but makes use of the conditional expectation method. The details python code is described in this Colab Notebook. The mathematics that lead to this estimator is described in this chapter:

Note that we want to know thanks to the back-door theorem:

$$P(y|do(T=t)) = \sum_{z} P(y|t,z)P(z) \tag{5}$$

but we are after the causal effect of T on Y, so we are after $\mathbb{E}(Y \mid do(T = 1)) - \mathbb{E}(Y \mid do(T = 0))$, so using the backdoor theorem we can rewrite this to:

$$\mathbb{E}(Y \mid do(T=1)) - \mathbb{E}(Y \mid do(T=0)) = \mathbb{E}_Z(\mathbb{E}(Y \mid T=1, Z)) - \mathbb{E}(Y \mid T=0, Z)).$$

So we can use any sort of model $\hat{\mu}$ to estimate the conditional expectation and then approximate the outer expectation by taking the sample mean:

$$\begin{aligned}\mathbb{E}(Y \mid do(T=1)) - \mathbb{E}(Y \mid do(T=0)) &= \mathbb{E}_Z(\mathbb{E}(Y \mid T=1, Z) - \mathbb{E}(Y \mid T=0, Z)) \\ &= \frac{1}{n}\sum_{i}^{n} \hat{\mu}(1, z_i) - \hat{\mu}(0, z_i).\end{aligned}$$

4.4 Doubly Robust Methods

Note that the correctness of both above estimation techniques depends on the correctness of the our choice of model $\hat{\mu}$ and $\hat{e}$. This is why some researchers have chosen to create a new estimation method called *doubly robust methods* that can correctly estimate $\hat{\tau}$ by using a combination of IPW and COM estimation techniques. The benefit is that the double robust methods estimates $\hat{\tau}$ correctly if either one of the chosen models $\hat{\mu}$ or $\hat{e}$ is correct.

4.5 S-Mint Approach of Jan Ernest and Peter Buhlmann

In [16], Buhlman and Ernest developed a marginal integration for nonparametric causal inference. The idea is as follows: first they calculate $\mathbb{E}(Y \mid do(T_i = t_i))$ by using the backdoor formula and adjustment set X_S. Because they are integrating over X_S, this allows for dimensionality reduction and they use an estimation technique that make use of marginal integration proposed by [17]. They first estimate regression of Y to X using adjustment set X_S and then average over the obtained estimates over X_S. They call this method S-mint and it achieves a major robustness result against misspecification. Note that this is in sharp contrast with the doubly robust methods proposed earlier.

5 Extraction Causal Graph

So far we have been working with predefined causal graphs and apply the do-calculus from there on. One of questions then remaining is how we get the causal graph. This is called *Causal Discovery.* Using the machinery of dependencies of graph structures defined in subsection 1.3 it is possible to extract causal graphs from raw data. There are variants of causal discovery and they all revolve around to what degree the assumptions are satisfied. We start by laying out those assumptions and then explain the algorithms that exists for various variants of causal discovery.

5.1 Assumptions and Definitions

We start by one of the assumptions that is the reverse of the global Markov assumption:

Assumption 6 *Faithfulness*

$$X \perp\!\!\!\perp_G Y \mid Z \Leftarrow X \perp\!\!\!\perp_P Y \mid Z.$$

This means that independence in the probability also implies independence in the graph. Of course, this is necessary when considering the observational data and converting that to a causal graph, but the drawback is that the assumption is much stronger. Some methods also imply another assumption called causal sufficiency:

Assumption 7 *Causal Sufficiency*
All confounders of variables have been taking into account.

Note that this is also a pretty strong assumption. Imagine dealing with observational data, then it is really hard to be sure that all the relevant variables have been taken into account. For this reason we will consider also methods that let go of this assumption. But first we introduce Markov equivalence classes. We go back to graph example of subsection 1.3 but now consider the case where we have only three variables and their observational data and the goal is to retrieve the causal graph structure from here.

Suppose that we have that all variables are unconditionally dependent, but conditional on only one of the variables the others are independent: $A_1 \perp\!\!\!\perp_P A_2 \mid A_3$. Then the results of the independence testing of subsection tell us that it either one of the graphs portrayed by Figure 9.

Because by testing for independence we can never be really sure which graphs it is, we say that all the graphs of Figure 9 belong to the same *Markov Equivalence Class*, because they entail the same conditional dependencies. Since we cannot direct the edges based on independence testing, in this scenario we can only derive the undirected graph or *skeleton* of the graph. The one thing we do know for certain is that we are not dealing with the graph structure as in

Figure 10, since we have seen that this one behaves very differently under independence testing. Unlike the other graph, this graph can be directed and then belongs to a different Markov equivalence class. Using independence testing, we can extract the causal graph up to its Markov equivalence class thanks to the following theorem by Pearl [9] and Frydenberg [11]:

Theorem 3 *Markov Equivalence Theorem*
Graphs are Markov equivalent if and only if they have the same skeleton and v-structures.

This means we can extract the partially directed acyclic graph (PDAG) based on independence testing. Note how the faithfulness theorem is central in this process. Two central algorithms to do this are PC and GES. We will discuss PC in detail:

5.2 PC algorithm for Causal Discovery

PC is named after Peter Spirtes and Clark Glymour [6] and is one of the central algorithms for causal discovery. It basically consists of three steps, skeleton identification, v-structure identification and rest-orientation where information from each step is saved to use in the following steps. We apply PC to extract the graph of Figure 11 to see how it works in practice.

5.2.1 Skeleton Identification

The first thing we do is start with the complete graph pointed out in Figure 12 and remove edges based on independence testing. Now we are going to start with unconditional independence testing. In this way, we break the relation between X and Y since they are unconditional independent and we can remove

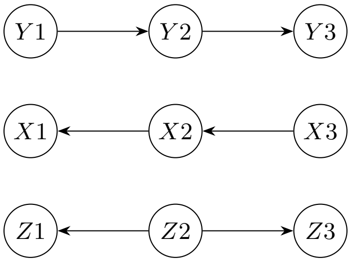


Figure 9: Markov Equivalent Graphs

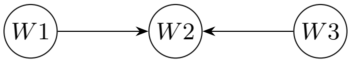


Figure 10: V-Structure

that arrow. Note that the arrow between X and Z remains since they are unconditional dependent. The algorithm then continues with conditional dependence testing of size 1. When we testing dependence between X and X conditional on W we will see that they are indeed independent and we can remove that arrow between the two nodes. The PC algorithm then continues with independence testing of conditional size 1 to remove the arrow between X and Z and to achieve a interesting result for dependence testing between X and Y conditional on W, namely they are dependent. The resulting graph from this procedure is Figure 14. In this case we have the result of the first step, but normally PC will continue.

5.2.2 V-structure Identification

In the previous step we accidentally came across a very interesting result: X and Y are unconditionally independent, but conditionally dependent on W. The only way this is possible is that X, W and Y form a v-structure, so we can now direct the arrows from X and Y to W. This step tests if there are any more v-structures based on the results of step 1 and the remaining conditional independence tests. In our example there are none, so we can continue to step 3.

5.2.3 Rest-orientation

Note that we have only one edge that is not yet directed: the edge between W and Z. Note that in a result of the PC algorithm, a partially directed acyclic graph, we are also dealing with undirected edges. However in the case of our example, we can direct that edge: if the edges was directed from Z to W, we would have a two more v-structures, but that contradicts our result from step 1 that X and Z are independent conditional W. Hence the arrow should be directed from W to Z and we achieve graph from Figure 11. This step aims to direct all arrows that contradict the v-structure conditional dependence.

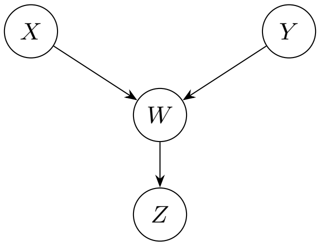


Figure 11: To be extracted graph

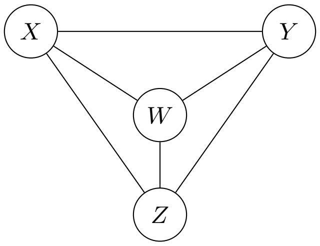


Figure 12: Markov Equivalent Graphs

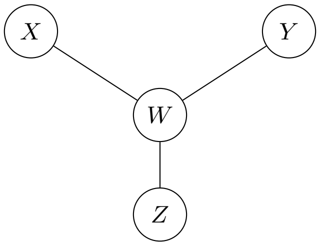


Figure 13: Graph obtained after step 1

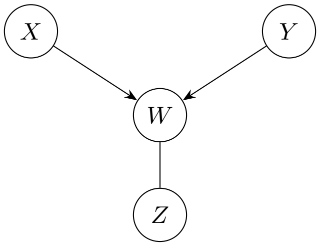


Figure 14: Graph obtained after step 2

5.3 Scoring algorithms and Notes

First we describe how scoring algorithms work and discuss one briefly, then we add some notes about causal discovery and special cases.

5.3.1 Scoring Methods and GES

Besides the PC-algorithm, we also have the GES algorithm from [8], which we will not explain in detail, but needs to be mentioned because it is a score based method. Score based methods basically assume one of the models from a markov equivalence class and fit it to the data. Then it scores the fit based on a scoring system (e.g. BIC score) and check for a model from a slightly different markov equivalence class. When the score from the second model is higher than it continues with the higher score model. The problem that arises is the enormous

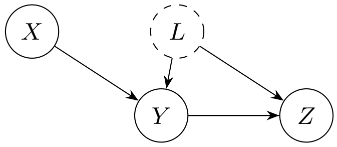


Figure 15: Causal graph with latent variables

search space that exists. This is where the GES (Greedy equivalence Search) comes in.

It has a forward and a backwards phase. The goal is to start with an empty model. The forward phase keeps keeps adding edges which improves the score most. Then, when no edges can be added that improve the score, the backwards phase starts with removing edges that improve the score most. When no edge can be removed that improves the score, the algorithm ends.

5.4 Complexity and Special Cases

One very important thing to mention is that many algorithms for causal discovery make use of conditional independence testing and this is very hard and requires many statistical data to achieve usable results [4] . That is one of the main practical problems of causal discovery.

Up to now we have only been able to do causal discovery up to its Markov equivalence class. However, when we assume additional conditions, we are able to add other edge orientation. This is the case when we assume linear non-Gaussian noise or when we assume nonlinear additive noise. We refer to [5] and [7] for the specifics for these cases in detail respectively.

As stated at the beginning of this chapter. There exist algorithms that can also deal with latent variables. We describe the specifics in the following subsection.

5.5 Causal discovery with latent variables

Because the causal sufficiency algorithm can be pretty strong assumption, there are many researchers that drop this assumption and consider the case where there can be missing confounders or latent variables. One of the most central in this discussion is the FCI algorithm (Fast Causal Inference) [6], which makes use of the CI algorithm(causal inference). We might discuss this algorithm to in the same way as we did with the PC algorithm, but this would call for a vast introduction of more graph-related concepts (induced paths, semi-colliders, definite discriminating path), so we save the details for later.

However, we illustrate the importance of such an algorithm by showing how PC extracts erroneous causal graphs when latent variables are involved. Also we show how the output of the FCI algorithm is slightly different than the PC algorithm, because it needs to account for latent variables.

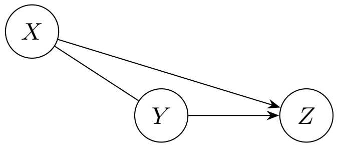


Figure 16: Result PC algorithm

5.5.1 Need for New Causal Discovery Algorithm

Suppose we are dealing with the graph as depicted in Figure 15. Imagine applying the PC algorithm to the observed variables here. In the first iteration of the first step, the PC algorithm checks for unconditional dependence and finds the complete graph containing X, Y and Z. In the second iteration, it checks for conditional dependence, in this case the dependence of Z and X conditioned on Y. Now, if we were not dealing with the latent variable Y, this would yield an independence, hence we would remove the arrow between X and Y, but because Y functions here as a collider, we receive dependency in the latent variable case. In this way PC would yield a complete graph as in Figure 16, which is false. Recall that the direction of the arrow from X to Y cannot be known. The FCI algorithm deals with this complication by removing edges in a later phase.

5.5.2 Output of FCI algorithm

Despite the fact that the algorithm will not be explained in full detail here, it is important to realize that the output of the FCI is different than that of the PC, because it needs to able to express the presence of latent variables. Therefore, we introduce PAGS. First observe that we can distinguish the vertices of a DAG now in observed variables O and latent variables L, hence we can write a DAG now as $G(O, L)$. It follows that Markov equivalence classes should be distinguished between O and L: O-Equiv($G(O, L)$) and L-Equiv($G(O, L)$). Then a PAG is defined to represent a DAG $G(O, L)$ or an O-equivalence class of $G(O, L)$ when all of the following hold

- The PAG has variables O.
- There is an edge between A and B in the PAG if and only if for every subset Z of O A, B A is d-connected to B conditional on Z in every DAG in O-Equiv($G(O, L)$).
- An edge between A and B in the PAG is oriented as $A \to B$ only if A is an ancestor of B in every DAG in O-Equiv($G(O, L)$).
- An edge between A and B in PAG is oriented as $A \leftrightarrow B$ only if B is not an ancestor of A and A is not an ancestor of B in any DAG in O-Equiv($G(O, L)$).

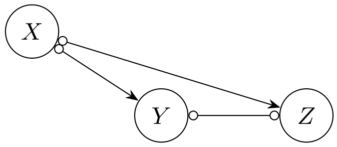


Figure 17: Result FCI algorithm

- An edge between A and B in PAG is oriented as $A \circ\!\!\rightarrow B$ only if B is not an ancestor of A in any DAG in O-Equiv$(G(O, L))$, and A is an ancestor of B in some but not all DAGs in O-Equiv$(G(O, L))$.
- An edge between A and B in PAG is oriented as $A \circ\!\!-\!\!\circ B$ only if B is not an ancestor of A in some but not all of the DAG in O-Equiv$(G(O, L))$, and A is an ancestor of B in some but not all DAGs in O-Equiv$(G(O, L))$.

The result of the FCI algorithm applied to graph in Figure 15 can then be displayed by the graph in Figure 17. Despite the erroneous arrow between X and Z, the FCI is able to retrieve the confounding relation between Y and Z.

5.5.3 Better algorithms within latent variables Causal Discovery

Even though the FCI can be seen as the foundation of causal discovery when latent variables are involved, there have been many progressions. Colombo et al [10] have developed a new algorithm called RFCI, because the FCI algorithm becomes computational infeasible for large graphs. Spirtes [12] has also made advances on the FCI algorithm, making it better to perform for small sample sizes. This algorithm is called GFCI.

Since PAG's represent the entire equivalence class, we are interested in a applying identification for certain elements of the equivalent classes, namely ADMG's. These acyclic directed mixed graphs are mixed graphs (containing directed and bidrected edges), but are DAG's when restricted to the directed edges. Note that the number of ADMG's grows exponential in the number $\circ\!\!\rightarrow$ edges in the PAG.

5.6 Alternative Approaches

There exists entirely different approaches as well. We are not go into detail of all of them, but the some characteristics of these approaches are described on a very general level.

5.6.1 Dormant Independence

Another component that plays a role in causal discovery is *dormant independence* [15]. Basically, dormant independence means conditional independencies that hold in the intervention distribution and can serve as a way to eliminate

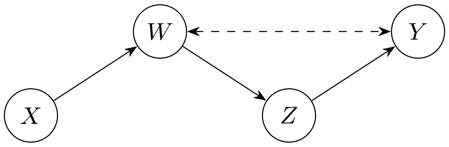


Figure 18: Dormant Independence Example.

extraneous edges in causal discovery. It is a concept that also comes back for factorizing intervention distributions.

We include an example from [15] to illustrate its relevance in causal discovery. If we look at the graph depicted by Figure 18 and we suppose we intervene on Z. We know that $P(x, y, w \mid do(z))$ is identifiable from $P(x, y, w, z)$. By d-separation of X and Y in the intervention distribution, we know that $P(y \mid x, do(z)) = P(y \mid do(z))$. But we also know that $P(y \mid x, do(z)) = \sum_w P(y \mid z, w, x)P(w \mid x)$. This then means that the expression only depends on y and z. In terms of causal discovery, this means that we can exclude the existence of an edge between X and Y in the original causal graph.

5.6.2 Inflation Technique

Another technique in causal discovery is called the inflation technique created by Spekkens et al [19], They focus on the following.: in the simplest case - when not considering latent variables- the algorithm is used to determine a verdict if compatibility holds if and only if the distribution is Markov with respect to the DAG, but the inflation technique is also able to determine compatibility in case of presence of latent variables. It is able to identify causal structures where other techniques fail. They describe their technique as follows:

"For a given causal structure under consideration, one can construct many new causal structures, termed inflation of this causal structure. An inflation duplicates one or more of the nodes of the original causal structure, while mirroring the form of the subgraph describing each node's ancestry. Furthermore, the causal parameters that one adds to the inflated causal structure mirror those of the original causal structure. We show that if marginal distributions on certain subsets of the observed variables in the original causal structure are compatible with the original causal structure, then the same marginal distributions on certain copies of those subsets in the inflated causal structure are compatible with the inflated causal structure. Similarly, we show that any necessary condition for compatibility of such distributions with the inflated causal structure translates into a necessary condition for compatibility with the original causal structure. Thus, applying standard techniques for deriving causal compatibility inequalities to the inflated causal structure typically results in new causal compatibility inequalities for the original causal structure." [19]

It is important to note that these techniques have been derived from quantum causality, which I may devote a chapter to at a later moment.

5.6.3 Invariant Causal Prediction

Buhlman et al created causal discovery by exploiting the invariance of a prediction under a causal model [21] [22]. Invariance is what we have described as modularity in the preliminaries. Given different experimental settings (for example various interventions) they collect all models that do show invariance in their predictive accuracy across settings and interventions. The causal model will be a member of this set of models with high probability and this gives rise to the introduction of confidence intervals. The call their algorithm Invariant Causal Prediction (ICP) and it is available in R here. Closer investigation of this theory can be executed later.

6 Identification and Factorization of Interventional Distributions

The completeness of do-calculus led to the famous algorithm ID, specified by Pearl ans Shpitser. This algorithm can identify interventions distributions if possible and, if not possible, state their unidentifiability. They make use of the graph concepts C-components and hedges to prove unidentifiability. The algorithm is already implemented in some Python packages (Dowhy). Despite the fact that the algorithm is complete in the sense that it can identify all causal effects, it is not able to provide a computational efficient scheme to calculate these interventional distributions. This is where the research of Shiptser [14] comes in to play. His new algorithm (EID) achieves some measure of computational efficiency by exploiting conditional and post-truncation independence constraints embedded in r-factorization. Fundamental in his research is the concept of r-factorization. Because this is an advanced mathematical component, [13] provides a more accessible entrance to factorization components. He did previous research that underlie the ideas of Shpitser. We first discuss the general idea of the ID algorithm and the concepts it relies on. Then, the shortcomings of the ID algorithms are discussed and how the EID algorithm fits into that. This link of a conference with Shpitser offers an accessible entrance to the EID algorithm with many examples.

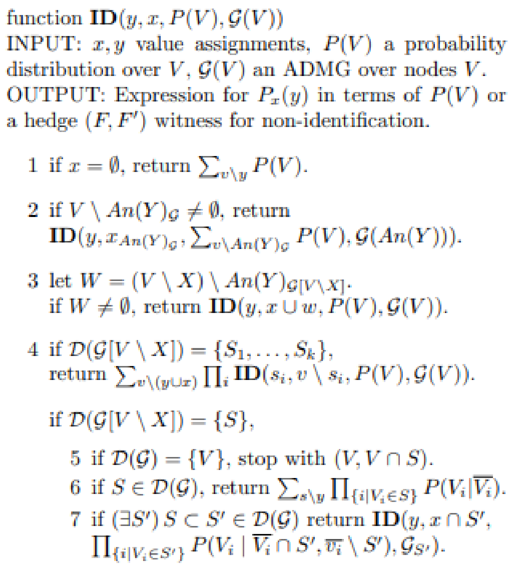
6.1 ID Algorithm

The ID algorithm is used to identify causal effects for ADMG's (the concept ADMG is explained in the chapter of Causal Discovery). Hence it takes as input an ADMG $G(V)$, a distribution $P(V)$, a intervention $X \in V$ and outcome $Y \in V$. It returns the intervention distribution $P(V \mid do(X)$ in terms of $P(V)$ or it fails. The notes about his algorithm here are derived from Shpitser [14], the algorithm is given in Figure 19 and the global idea of the algorithm is as follows:

First, the algorithm eliminates any irrelevant variables by intervening on them with arbitrary values (line 3). Second, the algorithm splits the problem into subproblems (line 4). Third, the algorithm marginalizes out some variables such that the margin that is left is an ancestral set in the graph resulting from interventions (line 2). And finally, the algorithm "truncates out" some variables in situations where the DAG truncation formula applies to a particular intervention (lines 6 and 7). When the algorithm fails (line 5), it returns a witness for this failure, called a 'hedge'.

For a complete example of how the algorithm works or fails see this link. Rather, we explain the witness of failure, which is called the hedge. Therefore we have to introduce two components:

Definition 3 *C-Forest: Let R be a subset of nodes in an ADMG G. Then a set F is called an R-rooted C-forest if $R \subset F$, F is a $\leftrightarrow$connected set, and every*



```
function $\mathbf{ID}(y, x, P(V), \mathcal{G}(V))$
INPUT: $x, y$ value assignments, $P(V)$ a probability
distribution over $V$, $\mathcal{G}(V)$ an ADMG over nodes $V$.
OUTPUT: Expression for $P_x(y)$ in terms of $P(V)$ or
a hedge $(F, F')$ witness for non-identification.

  1 if $x = \emptyset$, return $\sum_{v \setminus y} P(V)$.

  2 if $V \setminus An(Y)_{\mathcal{G}} \neq \emptyset$, return
    $\mathbf{ID}(y, x_{An(Y)_{\mathcal{G}}}, \sum_{v \setminus An(Y)_{\mathcal{G}}} P(V), \mathcal{G}(An(Y)))$.

  3 let $W = (V \setminus X) \setminus An(Y)_{\mathcal{G}[V \setminus X]}$.
    if $W \neq \emptyset$, return $\mathbf{ID}(y, x \cup w, P(V), \mathcal{G}(V))$.

  4 if $\mathcal{D}(\mathcal{G}[V \setminus X]) = \{S_1, \ldots, S_k\}$,
    return $\sum_{v \setminus (y \cup x)} \prod_i \mathbf{ID}(s_i, v \setminus s_i, P(V), \mathcal{G}(V))$.

    if $\mathcal{D}(\mathcal{G}[V \setminus X]) = \{S\}$,

      5 if $\mathcal{D}(\mathcal{G}) = \{V\}$, stop with $(V, V \cap S)$.
      6 if $S \in \mathcal{D}(\mathcal{G})$, return $\sum_{s \setminus y} \prod_{\{i|V_i \in S\}} P(V_i|\overline{V_i})$.
      7 if $(\exists S')\, S \subset S' \in \mathcal{D}(\mathcal{G})$ return $\mathbf{ID}(y, x \cap S',$
        $\prod_{\{i|V_i \in S'\}} P(V_i \mid \overline{V_i} \cap S', \overline{v_i} \setminus S'), \mathcal{G}_{S'})$.
```

Figure 19: ID Algorithm

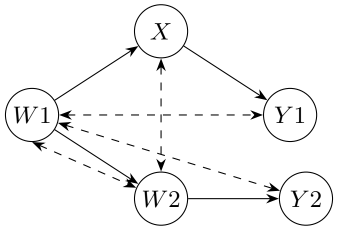


Figure 20: Hedge: because $F, F\backslash\{X\}$ with $F = G\backslash\{e\}$ and $e = \{W1, X\}$ are C-forests.

node in F has a directed path to a node in R with every element on the path also in F.

Definition 4 *Hedge Let X, Y be sets of variables in G. Let F, F' be R-rooted C-forests such that $F \cap X = \emptyset$, $F' \cap X = \emptyset$, $F' \subset F$ and $R \subset An(Y)_{G_{\overline{X}}}$ Then F, F' form a hedge for $P(y \mid do(x))$ in G.*

In Figure 20 and 21 are two examples: one is a hedge and one is not a hedge becauses it misses a path to the root node. In [2] Pearl proves that a hedge is a witness for unidentifiability. This is called the *hedge criterion*.

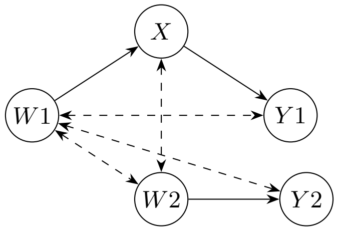


Figure 21: No Hedge: $F, F\backslash\{X\}$ with $F = G\backslash\{e\}$ and $e = \{W1, X\}$ are no C-forests, because there is no path from $W1$ to a root node.

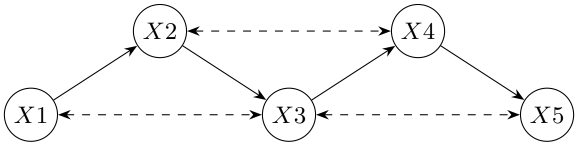


Figure 22: Example of application ID algorithm.

6.2 EID Algorithm

By showing the output of the ID algorithm for a graph in Figure 22, we show that there is a need for a better algorithm of ID. When we apply algorithm ID to graph in Figure 22

In this example, ID returns the following for the interventional expression $P(x_5 \mid do(x_3))$:

$$P(X_5 \mid do(X_3)) = \sum_{x'_4} (\sum_{x'_2} P(x_4 \mid x_3, x'_2, \overline{x_1}) P(x'_2 \mid \overline{x_1})) \times \quad (6)$$

$$(\sum_{x'_1, x'_3} P(x_5 \mid x'_4, x'_3, \overline{x_2}, x'_1) P(x'_3 \mid \overline{x_2}, x'_1) P(x'_1)) \quad (7)$$

Hence we can write the interventional distribution completely in the shape of conditional distributions. However, when we examine the terms we yield more closely, we see that in this quite simple ADMG we have a term $P(x_5 \mid x'_4, x'_3, \overline{x_2}, x'_1)$. Note that in the most favorable case, when all the variables are binary, this already yields 32 different conditional distributions to calculate. When ADMG's get more complex and the variables do not take only binary values, this conditional distribution becomes intractable. That is why we resort to [14] for an algorithm that not only identifies the interventional distribution, but also factorizes it in a advantageous way. In order to do this properly, we first discuss earlier factorization strategies developed by [13].

6.2.1 Preliminaries Factorization

There are more optimal ways of factorization, but we first have to introduce some additional graph concepts:

Definition 5 *A Markov blanket of a random variable Y in a random variable set $\mathcal{S} = \{X_1, \dots, X_n\}$ is any subset $\mathcal{S}_1$ of $\mathcal{S}$, conditioned on which other variables are independent with Y: $Y \perp\!\!\!\perp \mathcal{S} \backslash \mathcal{S}_1 \mid \mathcal{S}_1$.*

It means that $\mathcal{S}_1$ contains at least all the information one needs to infer Y, where the variables in $\mathcal{S} \backslash \mathcal{S}_1$ are redundant.

Furthermore, the set A is said to be *ancestrally closed* if $x \to \cdots \to a$ implies that $x \in A$, hence

$$\mathcal{A}(\mathcal{G}) = \{a \mid an_{\mathcal{G}}(A) = a\}$$

is the set of ancestrally closed sets. A vertex set A has a *barren*

$$barren(A) = \{x \mid x \in A; de_{\mathcal{G}}(x) \cap A = \{x\}\}$$

A *district* for a vertex x is the connected component (C-component) of x in $\mathcal{G}$:

$$dis_{\mathcal{G}}(x) = \{y \mid x \leftrightarrow \cdots \leftrightarrow \text{ in } \mathcal{G} \text{ or } x = y\}$$

A district $D = dis_W(v)$ of a vertex v in a set W in $\mathcal{G}$ is said to be ancestrally closed if for every vertex v

$$D = dis_{an_{\mathcal{G}}(D)}(v)$$

hence the district of v in W does not get any bigger once we start adding the ancestral set of D to W. When this is the case for all districts in W, it is said to have *ancestral closed districts.*

The goal of factorization will be to rewrite the distribution $P(X_A)$ for a set A in terms of $P(X_H, X_T)$ where H will be referred to as the *head* and T will be referred to as the *tail* of the term. We now only consider pairs (H, T) with the following

- $H = barren(an_{\mathcal{G}}(H))$
- H forms a path-connected set in $\mathcal{G}_a n(H)$
- $T = (dis_{an(H)}(H) \backslash H) \cup pa(dis_{an(H)}(H))$

6.2.2 Richarsons Factorization

With the preliminaries introduced, Richardson [13] now states that one can factorize an ADMG as follows:

Theorem 4 *A Probability Distribution P obeys the global Markov property for $\mathcal{G}$ if and only if for every $A \in \mathcal{A}(\mathcal{G})$,*

$$P(X_A) = \prod_{H \in [A]_{\mathcal{G}}} p(X_H \mid X_{tail(H)}) \qquad (8)$$

where $[A]_{\mathcal{G}}$ denotes a partition of A into sets $\{H_1, \dots, H_k\} \subset \mathcal{H}(\mathcal{G})$, with $\mathcal{H}(\mathcal{G})$ the set of heads.

example here

6.3 PID Algorithm

Another simplification of probablistic espression in causal inference has been created by Tikka and Karvanen [18]. They eliminate symbolically unnecessary variables from these expressions by taking advantage of the structure of the underlying graphical model and their algorithm has been implmented in the R package causaleffect.

They also have integrated this approach to create an improved algorithm of the ID algorithm called the Pruning Identification Algorithm (PID). [20] The idea behind this is to prune the probabilistic expression to a less complex expression. It is implemented in R and is to be found here.

7 Bayesian Inference

Bayesian Inference plays a central role in causal inference. This is because causal inference relies heavily on Bayesian networks in which the dependencies are embedded. In causal inference as well as bayesian inference, we are frequently dealing with probabilities and posterior probabilities, which are often intractable depending on the graph structure. In this chapter we pose several methods as to how to deal with this posterior distribution. We can make a distinction between exact methods like Variable Elimination and Belief Propagation as well as approximate methods such as sampling methods, Markov Chain Monte Carlo and Variational Inference. Many concepts have been extracted Ermongroup website.

7.1 Variable Elimination

We start with an exact method called *variable elimination* and explain its usefulness by comparing it to naive calculation of certain probabilities in a chain graph. Then we introduce some concepts and the formal algorithm to lay out the complete variable elimination algorithm.

7.1.1 Example Chain Graph

Suppose we are dealing with the following *chain graph* in Figure 23. We marginalize out all of the variables to calculate the probability of $P(x_n)$ naively:

$$P(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} P(x_1, \dots, x_n)$$

Assuming every variable X_i can take k values, this operation would take $O(k^n)$ steps, but we can simplify this if we assume the Local Markov Assumption:

$$P(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} P(x_1) \prod_i P(x_i \mid (x_{i-1})).$$

We can then rewrite this by pushing the sums in the equation as deep as possible.

$$P(x_n) = \sum_{x_{n-1}} P(x_n \mid x_{n-1}) \sum_{x_{n-2}} P(x_{n-1} \mid (x_{n-2})) \cdots \sum_{x_1} P(x_2 \mid x_1) p(x_1).$$

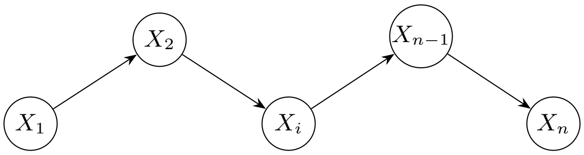


Figure 23: Chain Graph

For simplicity reason we can rewrite every sum component into $\tau(x_2) = \sum_{x_1} p(x_2 \mid x_1)p(x_1)$ and $\tau(x_3) = \sum_{x_2} p(x_3 \mid x_2)\tau(x_1)$, so that we yield $\tau(x_{n-1}) = p(x_n)$.

This takes $O(k^2)$ steps for every calculation of $\tau(x_i)$ and hence $O(nk^2)$ in total. Even though this is much better then the naive marginalization of $O(k^n)$ steps, much is due to the favorable structure of the graph. Note that when more arrows point to one node, we would yield $O(k^{a-1})$ per $\tau(x_i)$ calculation with a the number of edges pointing to node x_i.

7.1.2 Factors and Algorithm

Recall the Bayesian Network Factorization in Section 1:

$$P(x_1, \dots x_n) = P(x_1) \prod_i P(x_i \mid \mathrm{pa}(x_i)).$$

We will rewrite the conditional probability distributions into factors: $\phi(x_i) = P(x_i \mid \mathrm{pa}(x_i))$. Now given ordering $i = 1, \dots, n$ the variable elimination algorithm can informally be stated as follows: For $i = 1, \dots, n$

- Group all factor containing X_i.
- Multiply those factors.
- Marginalize out X_i.
- Replace the factors by the new factor τ

The order that we give is essential in evaluating the efficiency of the algorithm. When we return to the chain graph, then the natural order by the DAG was necessary to first factorize $\tau(x_2) = \sum_{x_1} p(x_2 \mid x_1)p(x_1)$ and then factorize $\tau(x_3) = \sum_{x_2} p(x_3 \mid x_2)\tau(x_1)$. However sometimes the ordering is not obvious and the ordering can play an essential role in the computational complexity. Finding the best ordering is an NP-hard problem. This is especially the case in Markov Random Fields instead of DAGS.

7.2 Belief Propagation, Join Tree

To be written

7.3 Sampling Methods

To be written (should include (clipped) importance sampling since it is proposed in methods [23] and [24].

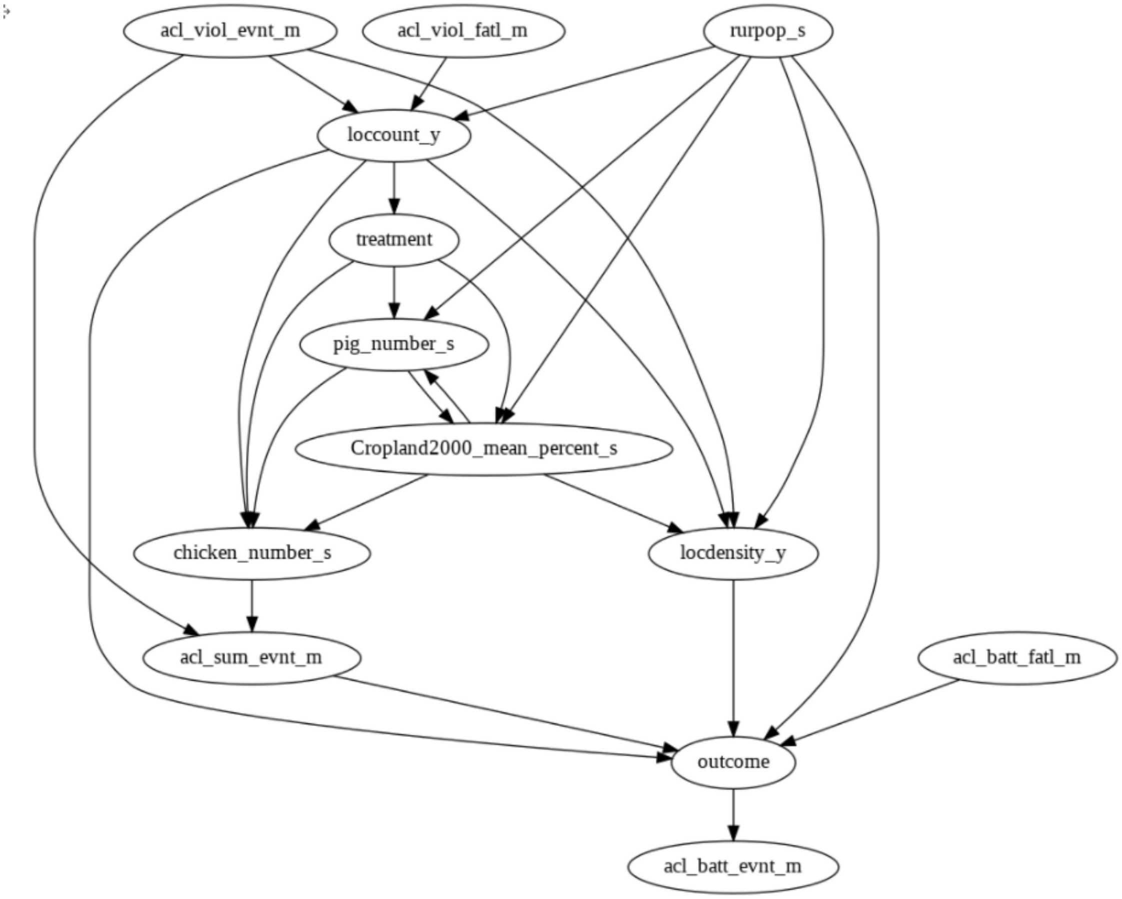


Figure 24: Causal Inference with multiple variables

7.4 Markov Chain Monte Carlo

7.5 Variational Inference

8 Optimization problem

Having introduced the do-calculus and a simple application, we can start how generalizing this idea brings us to the optimization exercise that lies ahead: consider the following graph in Figure 24 were the treatment and outcome variables have already been specified. The causal effect of the treatment variable can be defined as the causal effect

$$\mathbb{E}(Y \mid do(T=1)) - \mathbb{E}(Y \mid do(T=0))$$

for treatment variable T and outcome variable Y if T is binary. And we define the (marginal) causal effect of treatment T on outcome variable Y when T continuous as:

$$\frac{\mathbb{E}(Y \mid do(t)) - \mathbb{E}(Y \mid do(t - \Delta t))}{\Delta t}.$$

Thanks to the do-calculus, if the outcome is identifiable given the treatment, we can rewrite this expression containing a do-operator to an expression not containing a do-operator. Once we have such an expression, we can use statistical techniques to approximate this expression. In the example of the previous chapter, the objective was obvious, because we have one treatment variable and it was binary. But considering the graph in Figure 24, we are not merely interested in the causal effect of one binary variable on the outcome, but we are interested in knowing what variables to intervene on and what to set as the intervention value in order to yields the most desirable outcome given a maximal number of variables to intervene on. Mathematically, this means we are interested in:

$$\max_{S,t_i} \mathbb{E}(Y \mid \bigcup_{T_i \in S} do(T_i = t_i))$$

where $\mid S \mid \leq n$.

Let for simplicity first consider the case were we are interested in finding the intervention value that maximizes the expected outcome given a certain intervention T:

$$\max_{t} \mathbb{E}(Y \mid do(T = t))$$

.

Then we can consider multiple interventions and find the intervention values that maximize the expected outcome:

$$\max_{t_1,\ldots,t_n} \mathbb{E}(Y \mid do(T_1 = t_1, \ldots, T_n = t_n))$$

.

Now we consider the case were we do not only have to find the intervention value, but also the variable to intervene on, considering at most one variable to intervene on:

$$\max_{i,t_i} \mathbb{E}(Y \mid do(T_i = t_i))$$

Of course, the goal is to find multiple intervention variables and the intervention values that maximizes our objective with the constraint of only n interventions:

$$\max_{S,t_i} \mathbb{E}(Y \mid \bigcup_{T_i \in S} do(T_i = t_i))$$

where $\mid S \mid \leq n$.

8.1 Possible Approaches

In this chapter, I collect useful papers that may or may not pose valuable approaches to the optimization problem formulated in the previous chapter and then outline their proposed methods. Note that the previous chapters can be seen as providing the preliminaries of this chapter. Chapter 2 and 3 offer the preliminaries for causal graphs and the do-operator. Chapter 4 offers applications and estimation techniques for causal estimation. Chapter 5 describes the causal discovery exercise and state of the art algorithms that exist within this context. Chapter 6 describes identification algorithms and some factorization techniques they make us of. Chapter 7 serves as Bayesian Inference techniques that are of paramount importance for causal estimation.

It turns out that there exists a vast amount of research investigating a strategy of interventions. This is know as the *Structural Causal Bandits*, a mixture of structural causal models and Multi-armed bandit problem. The Multi-armed bandit is suitable for the strategy of interventions, because in the setting of a strategy of intervention, playing an arm corresponds to intervening on a set of variables and setting them to specific values. We first describe the Multi-armed bandit problem and its variations and then propose a couple of possible approaches of for the application of structural causal bandits in different settings.

8.1.1 Multi-armed bandit

We try to keep the same notation as in the proposed methods in the discussion of approaches within the context of strategy of interventions:

Definition 6 *Suppose we have a finite number of arms denoted by $i = 1, \dots, K$ with associated reward distributions $\{\mathcal{D}_1, \dots, \mathcal{D}_k\}$ and associated mean values $\{\mu_1, \dots, \mu_k\}$. At each timestep $t = 1, \dots, T$, we take action on one arm and receive reward r. $\mathcal{A}$ is the set of actions. The value of action $a \in \mathcal{A}$ is the expected reward μ_a. The best mean reward is denoted by $\mu^* = \max_{a \in \mathcal{A}} \mu(a)$. After T rounds, the algorithms returns an estimate of the optimal arm $\hat{a}_T^*$. We define the simple regret R_T of at round T is the difference between the best mean reward of the recommended arm $\hat{a}_T^*$ at time T.*

$$R_T = \mu^* - \mathbb{E}[\mu_{\hat{a}_T^*}]$$

Although in many application of the multi-armed bandit, the goal is the minimize the cumulative regret, we will see that in the context of the strategy of interventions, the simple regret is more apt. There are several algorithms as how to tackle this problem. The simplest ones are the explore-first algorithms.

8.1.2 Solutions to Multi-armed bandit problems

To be written if necessary

8.1.3 Contextual multi-armed bandit

Another important variant of the multi-armed bandit is the contextual multi armed bandit where the algorithm first observes context x_i before choosing arm a_i in round i. In this case the reward r_t in each round t depends on the action a_t as well as the context x_t. The expected reward of action a and context x is denoted by $\mu_{a,x}$.

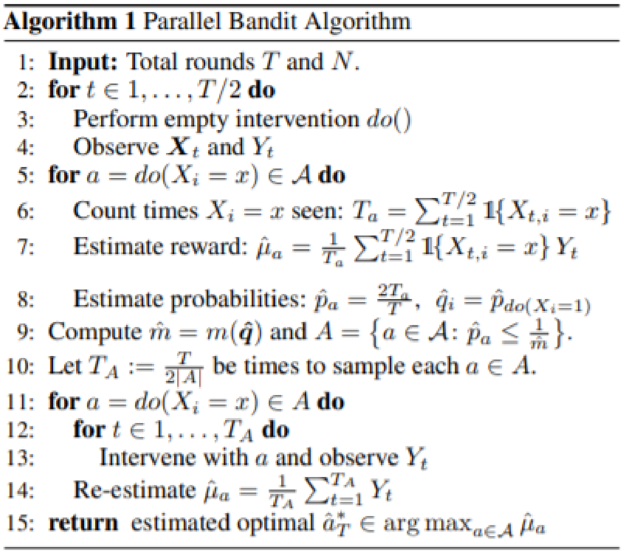
8.1.4 Classic Approach of Lattimore

The first to create a strategy of interventions was Lattimore in his paper and he created the structural causal bandit problems [23]. It was unique in the sense that the classical multi-arm bandit as well as the contextual arm bandit problem were enhanced to fit for interventions. The key difference from the classical bandit problem is that in the causal bandit problem there is additional information about unintervened distribution and the difference from the contextual bandit problem is that additional information is only gained after intervening on a variable. They propose a new algorithm that includes this causal intervened feedback and find bounds on the regret. Sen has built upon the research of Lattimore to improve the algorithm, so they now provide better bounds for the regret [24]. He makes uses of clipped importance sampling. There is even more recent research that even improve this bounds [25].

Because Lattimore was the founder of framing the problem as a multi-arm bandit problem, we will discuss his problem formulation and solution algorithm in more detail by providing useful examples. Lattimore makes a distinction between the parallel bandit solution and the general bandit solution, but we first describe the problem setup by reframing the classic multi-arm bandit problem as a causal bandit problem:

Definition 7 *Suppose we have a directed acyclic graph G containing random variables $X_1, \cdots, X_N$. Let P be a distribution function over the random variables. For simplicity we assume that every random variable can only take a finite number of values. An edge from X_i to X_j means that a change in X_i leads to a change in X_j. We are given a set of allowed actions $\mathcal{A}$ corresponding to interventions, $a \in \mathcal{A}$ with $a = do(X = x)$, where an intervention is evaluated in random variable Y by the expected reward $\mu_a := \mathbb{E}[Y \mid do(X = x)]$ and the optimal reward given by $\mu* = \max_{a \in \mathcal{A}} \mu_a$. The causal bandit problem lasts T round where in every round t, action $a_t = do(X_t = x_t) \in \mathcal{A}$ and observes sample values for all non-intervened variables drawn from $P(X_t \mid do(X_t = x_t))$ including reward $Y_t \in \{0, 1\}$. After T observations the algorithm outputs an estimate for the optimal action $\hat{a}^*_T \in \mathcal{A}$.*

Parallel Bandit First we consider the case where we all the random variables are independent causes of the reward variable Y. We consider interventions of size 0, the empty intervention $do()$, an of size 1, $do(x_i = j)$, where we take j to be binary for simplicity. Every round corresponds then to doing the empty



```
Algorithm 1 Parallel Bandit Algorithm
Input: Total rounds $T$ and $N$.
for $t \in 1, \dots, T/2$ do
    Perform empty intervention $do()$
    Observe $\boldsymbol{X}_t$ and $Y_t$
for $a = do(X_i = x) \in \mathcal{A}$ do
    Count times $X_i = x$ seen: $T_a = \sum_{t=1}^{T/2} \mathbb{1}\{X_{t,i} = x\}$
    Estimate reward: $\hat{\mu}_a = \frac{1}{T_a} \sum_{t=1}^{T/2} \mathbb{1}\{X_{t,i} = x\} Y_t$
    Estimate probabilities: $\hat{p}_a = \frac{2T_a}{T}$, $\hat{q}_i = \hat{p}_{do(X_i=1)}$
Compute $\hat{m} = m(\hat{\boldsymbol{q}})$ and $A = \left\{a \in \mathcal{A} \colon \hat{p}_a \leq \frac{1}{\hat{m}}\right\}$.
Let $T_A := \frac{T}{2|A|}$ be times to sample each $a \in A$.
for $a = do(X_i = x) \in A$ do
    for $t \in 1, \dots, T_A$ do
        Intervene with $a$ and observe $Y_t$
    Re-estimate $\hat{\mu}_a = \frac{1}{T_A} \sum_{t=1}^{T_A} Y_t$
return estimated optimal $\hat{a}^*_T \in \arg\max_{a \in \mathcal{A}} \hat{\mu}_a$
```

Figure 25: Parallel Bandit Algorithm

intervention or intervention of size 1 and then observing the variables to determine the distribution of the rewards. When we do the empty-intervention, this means we assume X_i $Bernoulli(q_i)$ where $q = (q_1, \dots, q_N) \in [0,1]^N$ and $q_i = P(X_i = 1)$. We first outline the complete algorithm and then discuss an example to intuitively describe how it works.

Algorithm In the first $T/2$ rounds, we do the empty intervention $do()$ to collect observational data. Since there are no incoming arrows in any of the causal graphs, we know that $P(Y \mid do(X_i = j)) = P(Y \mid X_i = j)$, which gives us the opportunity to evaluate actions $do(X_i = j)$ when $P(X_i = j)$ is large. For when this is not the case, we use the remaining $T/2$ rounds to evaluatie the rewards of these actions. This depends heavily on q and how many q_i or $1 - q_i$ are small. Let $\tau \in \{2, \dots, N\}$ let $I_\tau = \{i : min\{q_i, 1 - q_i\} < \frac{1}{\tau}\}$ and we define $m(q) = min\{\tau : |I_\tau| \leq \tau\}$. Note that $m(q)$ is used to define the 'smallness' of q: when $q = (0.5, \dots, 0.5)$, $m(q) = 2$ and when $q = (0, \dots, 0)$, $m(q) = N$. The algorithm is given by Figure 25.

Example Suppose we have $T = 4$ and $N = 2$ and the set of considered actions A consists of intervening on x_1 and X_2 with the value set to $\{0, 1\}$, hence 4 actions in total. We use made-up observational results that best explain the algorithm.

We start with the first for loop for $t = 1, 2.$ and we perform the empty intervention twice to observe for $t = 1$: $X_1 = 0$ and $X_2 = 1$ and $Y = 0$ and

$X_1 = 1$ and $X_2 = 1$ and $Y = 1$. In the second loop, we loop over all the actions:

For $a = do(x_1 = 0)$, we yield $T_a = \sum_{t=1}^{2} \mathbb{1}\{x_t, 1 = 0\} = 1$, which is the occurrence. The estimated reward is $\hat{\mu}_a = \frac{1}{T_a} \sum_{t=1}^{2} \mathbb{1}\{X + t, i = x\} Y_t = \frac{1}{1} 0 = 0$. Then $\hat{p}_a = \frac{1}{2}$ and $q = \hat{p}_{do(X_1)=1)=\frac{1}{2}}$. We can do the same for the other actions to eventually receive final $q = (\frac{1}{2}, 1)$, $m(q) = 2$ and $A = \{a \in \mathcal{A} : \hat{p}_a \leq \frac{1}{m} = \{do(X_1 = 1), do(X_1 = 0), do(X_2 = 0)\}$. We have $T_A = \frac{T}{2|A|} = \frac{2}{3}$ be times to sample each action. So because we have such a small example, we yield a fraction as sample size, but normally we would intervene with actions in A with sample size T_A to yield new, better estimates. we are returned with the highest μ of all interventions, so in our case that would be the μ corresponding with action $do(X_1 = 1)$.

General Bandit Very vague, trying to figure out what is going on here.

Sen's contribution In order to fully grasp the contribution of Sen, we should first make a distinction between soft (parametric) and hard (structural) interventions. This distinction is extensively discussed in Chapter 4. Sen only considers soft interventions where the conditional distribution relating $pa(V)$ and V is changed to $P_k(V \mid pa(V))$ for interventions k where $P_k(V \mid pa(V))$ is considered to be known. Because considering soft interventions can make the problem more dynamic, they limit themselves by only considering one variable V where the intervention takes place.

8.1.5 Contributions of Bareinboim

Bareinboim is a top researcher when it comes down to causality in general or strategy of interventions. He has made contributions in situation with allow for mixed policy interventions [30], unobserved confounders [26] [27], in the case where not all variables are manipulable [29] and in case the underlying causal model is not fully recognized [28].

References

[1] Pearl, J, (1995) Causal diagrams for empirical research: Rejoinder to Discussions of 'Causal diagrams for empirical research', In *Biometrika, Volume 82, Issue 4* Pages 702–710

[2] Shpitser, I and Pearl, J, (2006) Identification of Joint Interventional Distributions in Recursive SemiMarkovian Causal Models. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2.* Pages 1219–1226

[3] Neal, B (2020) Introduction to Causal Inference. $https://www.bradyneal.com/Introduction_{t}o_{C}ausal_{I}nference - Dec17_{2}020 - Neal.pdf$

[4] Shah, R., Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *The Annals Of Statistics, 48(3). doi: 10.1214/19-aos1857*

[5] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. (2006) A Linear Non-Gaussian Acyclic Model for Causal Discovery. In: *Journal of Machine Learning Research 7.72*, pp. 2003–2030

[6] Spirtes, P., Glymour, C., Scheines, R. (2000). Causation, prediction, and search. Cambridge, Mass.: MIT Press.

[7] Kun Zhang and Aapo Hyvärinen. (2009) On the Identifiability of the Post-Nonlinear Causal Model. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence.* UAI '09. Montreal, Quebec, Canada: AUAI Press, 2009, pp. 647–655

[8] Chickering, David Maxwell. (2002) Optimal structure identification with greedy search. In *Jorunal of machine learning research 3*

[9] Thomas Verma and Judea Pearl. (1990) Equivalence and Synthesis of Causal Models. In: *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence. UAI '90. USA: Elsevier Science Inc.*

[10] Colombo, D., Maathuis, M., Kalisch, M., Richardson, T. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. In *The Annals Of Statistics, 40(1). doi: 10.1214/11-aos940*

[11] Morten Frydenberg. (1990) The Chain Graph Markov Property. In: *Scandinavian Journal of Statistics 17.4*

[12] JM, O., P, S., J, R. (2021). A Hybrid Causal Search Algorithm for Latent Variable Models. Retrieved 11 August 2021, from https://pubmed.ncbi.nlm.nih.gov/28239434/

[13] T.S. Richardson. (2003) Markov properties for acyclic directed mixed graphs. In *Scand. J. Statist., 30:145–157*

[14] I. Shpitser and J. M. Robins (2011). An efficient algorithm for computing interventional distributions in latent variable causal models. In *27th Conference on Uncertainty in Artificial Intelligence (UAI-11).* AUAI Press

[15] I. Shpitser and J. Pearl. (2008) Dormant independence. In *Twenty Third National Conference on Artificial Intelligence* (AAAI-08), pages 1081–1087. AAAI Press

[16] Ernest, J., Buehlmann, P. (2015). Marginal integration for nonparametric causal inference. In *Electronic Journal of Statistics, 9(2), 3155–3194.* https://doi.org/10.1214/15-EJS1075

[17] Linton, O., Nielsen, J. P. (1995). A kernel method of estimating structured nonparametric regression based on marginal integration. In *Biometrika*, 82(1), 93–100. https://doi.org/10.1093/biomet/82.1.93

[18] Tikka, S., Karvanen, J. (2017). Simplifying Probabilistic Expressions in Causal Inference. Journal of Machine Learning Research, 18.

[19] Wolfe, E., Spekkens, R. W., Fritz, T. (2019). The Inflation Technique for Causal Inference with Latent Variables. In *Journal of Causal Inference*, 7(2). https://doi.org/10.1515/jci-2017-0020

[20] Tikka, S., Karvanen, J. (2017). Enhancing Identification of Causal Effects by Pruning. J. Mach. Learn. Res., 18, 194-1.

[21] Peters, J., Bühlmann, P., Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. In Journal of the Royal Statistical Society. Series B, Statistical Methodology, 78(5), 947–1012. https://doi.org/10.1111/rssb.12167

[22] Meinshausen, N., Hauser, A., Mooij, J. ., Peters, J., Versteeg, P., Bühlmann, P. (2016). Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences - PNAS*, 113(27), 7361–7368. https://doi.org/10.1073/pnas.1510493113

[23] Lattimore, F., Lattimore, T., Reid, M. D. (2016). Causal bandits: Learning good interventions via causal inference. arXiv preprint arXiv:1606.03203.

[24] Sen, R., Shanmugam, K., Dimakis, A. G., Shakkottai, S. (2017, July). Identifying best interventions through online importance sampling. In International Conference on Machine Learning (pp. 3057-3066). PMLR.

[25] Lu, Y., Meisami, A., Tewari, A., Yan, W. (2020, August). Regret analysis of bandit problems with causal background knowledge. In Conference on Uncertainty in Artificial Intelligence (pp. 141-150). PMLR.

[26] Bareinboim, E., Forney, A., Pearl, J. (2015). Bandits with unobserved confounders: A causal approach. Advances in Neural Information Processing Systems, 28, 1342-1350.

[27] Zhang, J., Bareinboim, E. (2017, May). Transfer learning in multi-armed bandit: a causal approach. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (pp. 1778-1780).

[28] Lee, S., Bareinboim, E. (2018). Structural causal bandits: where to intervene?. Advances in Neural Information Processing Systems 31, 31.

[29] Lee, S., Bareinboim, E. (2019, July). Structural causal bandits with non-manipulable variables. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 4164-4172).

[30] Lee, S., Bareinboim, E. (2020). Characterizing optimal mixed policies: Where to intervene and what to observe. Advances in neural information processing systems, 33.

[31] Eberhardt, F., Scheines, R. (2007). Interventions and causal inference. Philosophy of science, 74(5), 981-995.

[32] Correa, J., Bareinboim, E. (2020, April). A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 06, pp. 10093-10100).