# From Data to Causal Structure: Representation & Search

**Master's Thesis Report**

Jean-Philippe Pellet

September 19th, 2006

ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
Laboratory of Computational Neurosciences

IBM ZURICH RESEARCH LABORATORY
Business Optimization Group

*Supervised by*
Prof. Dr. Wulfram Gerstner
Dr. André Elisseeff

*The law of causality, I believe, like much that passes muster among philosophers, is a relic of a bygone age, surviving, like the monarchy, only because it is erroneously supposed to do no harm.*

—BERTRAND RUSSELL (1913)

*Now it is an ironical but familiar fact that though the business of science is describable in unscientific language as the discovery of cause, the notion of cause itself has no firm place in science. The disappearance of causal terminology from the jargon of one branch of science and another has seemed to mark the progress in understanding of the branches concerned.*

—W. V. O. QUINE (1954)

*Development of Western science is based on two great achievements: the invention of the formal logical system (in Euclidian geometry) by Greek philosophers, and the discovery of the possibility to find out causal relationships by systematic experiment (during the Renaissance).*

—ALBERT EINSTEIN (1953)

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor and mentor, André Elisseeff, who often had to tolerate my lack of knowledge, a lot of questions whose answers were obvious, and a few unexpected off-topic remarks after a moment of fake reflection.

I would also like to thank my friend and colleague Ulf Nielsen for the fruitful discussions we had, be it while trying to understand some algorithmic detail or while discussing the causality problem or some other (possibly related) topic. (The first example describing V-structures was inspired by him.)

I have to mention my long-time close friend, classmate and colleague Sébastien "Tall One" Cuendet and thank him for the fertile, late-night IM debates between Berkeley and Zurich on machine learning and the meaning of the derivative of the life–value function.

Many thanks to my family, who had to put up with me for the last weeks before the deadline, for useful input concerning layout and readability as well as linguistic corrections and valuable help with the printouts.

Finally, I am indebted to the whole Business Optimization Group at IBM Research, with whom working became a pleasure within a few days and eating together often ended in collapses of laughter. This is a debt I am looking forward to repaying.

# ABSTRACT

The knowledge of cause–effect relationships is the core of our understanding of a given system. However, deriving these relationships from data has been controversial for decades—generations of statisticians and philosophers simply dismissed the problem of unveiling causation by examining observational data.

In this study, we show how important it is to know the causal structure of a system. Considering the example of Simpson's paradox, a century-old misinterpretation of conditional probabilities, we explain how causal considerations are essential to assess the effect of interventions on a system.

We then review the recent developments in causality detection and show how direct cause–effect relationships can be formalized and represented. In certain cases and with certain assumptions, direct causation can be detected: we build on this to discuss several causal network construction algorithms and test them on toy data and on real-world retail data.



**Keywords:** causation, causal networks, graphical methods, Simpson's paradox, Bayesian networks, structural equation models, structure learning.

# NOTATION

| | |
|---|---|
| $X$ | Random variable, in this document simply called "variable" |
| $\mathcal{X}$ | Range of variable $X$ |
| $x$ | Specific instantiation of variable $X$; $x \in \mathcal{X}$ |
| $X_i$ | Variable indexed for some purpose |
| $\mathbf{V}$ | Set of variables *or* set of vertices |
| $\mathbf{v}$ | Specific instantiation of the set of variables $\mathbf{V}$ |
| $\theta$ | Parameter of some model |
| $\hat{\theta}$ | Some estimator of parameter $\theta$ |
| NAME | Variable with a longer name *or* named node in a graph |

| | |
|---|---|
| $\mathbf{X}$ | Matrix |
| $\mathbf{X}^T$ | Transposed matrix of matrix $\mathbf{X}$ |
| $\mathbf{X}^{-1}$ | Inverse matrix of matrix $\mathbf{X}$ |
| $(x_{ij})$ | Matrix where the $(i,j)$th entry is $x_{ij}$ |
| $x_{ij}$ | $(i,j)$th entry of matrix $\mathbf{X}$ |
| $\mathbf{I}$ | Identity matrix |

| | |
|---|---|
| $\mathbf{w}$ | Column vector |
| $\mathbf{w}_i$ | Column vector indexed for some purpose |
| $\mathbf{w}^T$ | Transposed (row) vector of column vector $\mathbf{w}$ |
| $w_i$ | $i$th element of vector $\mathbf{w}$ |
| $a$ | Scalar |

**Specific variables in this document:**

| | |
|---|---|
| $D$ | Dataset |
| $n$ | Number of variables in a specific context |
| $p$ | Number of samples in a specific context |
| $\mathcal{G}$ | Graph representing a dataset |

# Table of Contents

CHAPTER

1

# Introduction

DESCRIBING and understanding our environment not only with facts but with the help of causal considerations is commonplace. Looking at a few newspaper articles, we find headlines such as "Coffee as a health drink?" (Bakalar, 2006), "Deodorizers may cause reduction in lung function" (O'Connor, 2006), "A valuable drug, with side effects" (Graedon & Graedon, 2006), all hinting at the underlying search for causation. In economics, physics, social sciences, and biological sciences, among others, a lot of studies aim at explaining how a system works by pointing out the details of some cause–effect relationships.

The natural interpretation of most of the events occurring around us is causal, and lies in the implicit detection of causation. Finding explanations for a phenomenon is often associated with finding its causes, because knowing them and knowing how they affect the phenomenon of interest eventually lets us influence it in a predictable way, or, alternatively, understand why we cannot influence it.

Children learn to deal with causes and effects very young, as they evaluate whether crying long enough lets them get more attention from their parents, or how moving certain parts of their body helps them approach an object of interest. Learning causation seems natural, and is usually deduced from the repeated coincidence of several phenomena—which we will denote as "the detection of a significant correlation between two variables"—with the help of common sense, expert knowledge or additional experiments to determine which of two correlated variables causally influences the other.

But we all know the adage which says that "correlation is not causation." It is not because the growth of my hair is strongly correlated with the growth of my fingernails that the one causes the other: in this case, the correlation only reveals a spurious association, due to a possibly unobserved common cause. The difficulty of unveiling spurious associations and the problem of differentiating a cause from an effect with tools from traditional statistics have probably resulted in the fact that causality and causation detection are either absent or dismissed as irrelevant in most classical statistics textbooks today—the three opening quotes, the last of which is re-cited from Pearl (2000), are meant to underline this controversial aspect. Today, the possibly only recognized technique dealing with causation detection is the randomized control experiment. But no common unified language exists to mathematically describe causation.

Attempts were made, however. In this document, we want to explain how and under what assumptions correlation can be interpreted as causation, how we can formalize cause–effect relationships, and how to represent graphically and algebraically the causal structure of datasets. We are interested in detecting causation in repeatable events to better understand a system and help predict the effect of new policies or interventions on the system. This study aims at presenting the problem and its challenges in a single document. It describes the state of the art in causal structure detection and investigates alternative approaches.

In Chapter 2, we motivate the need for causal considerations with the puzzling example of Simpson's paradox, and underline the difference between observation and intervention. We then introduce the causation detection problem intuitively, review basic statistical tools, and determine what we can do and where the fundamental limitations are.

Building on this, Chapter 3 introduces more formally the concept of causal graphs, justifies their use as a graphical representation of causation, and mentions their relation to Bayesian networks and to Structural Equation Modeling. We then briefly review how they can be used to compute the effect of interventions.

Chapter 4 explains algorithms which build causal graphs from observational data. We review their assumptions, test them and detail the non-straightforward implementation steps.

In Chapter 5, we discuss alternative approaches to the causation problem by presenting an algorithm based on partial correlation. We also relate causality to the problem of feature selection.

The description of a practical example involving real-world retail data as well as results of experiments can be found in Chapter 6, where we apply our algorithm on a series of experimental scenarios.

We finally conclude in Chapter 7, wrapping up and reviewing the limitations of current methods, mentioning again outstanding issues.

CHAPTER

2

# Causality: Importance & Principles

*In this chapter, we explain the importance of causality and of the knowledge of the causal structure of a given system. We first explain the difference between* observing *that a variable has a certain value, and* setting *this variable to the same value. To underline this difference, we examine an illustration of Simpson's paradox and show that the causal structure helps interpret the data.*

*We then proceed to see why causality detection is not straightforward given observational data, and try to introduce intuitively the basic principles of structure construction with the help of conditional independence. We finally explain the limitations and remaining ambiguities of the interpretation of a set of conditional independence statements.*

## 2.1 Observation vs. Intervention

Detecting the cause–effect relationships is generally necessary to understand how a system reacts under different conditions. In most cases, a purely statistical analysis based on conditional independence is not enough. To illustrate this, we show that there can be major differences between *observing* that a variable is in a certain state (i.e., when conditioning on that variable), and *setting* this variable to the same state. Only if we know the causal structure can the effect of the latter be computed.

To that end, we present a century-old statistical puzzle, Simpson's paradox.

### 2.1.1 Motivating Example: Simpson's Paradox

Formally, Simpson's paradox, stated by Simpson (1951) and first described by Yule (1903), says that any statistical relation can be reversed by including more factors in the analysis. For instance, suppose a new kind of drug is developed and a study is conducted to know whether it is effective. It is mathematically plausible to find that the probability of recovering with the new drug is worse when conditioning on the gender of the patient, and also observe that the same probability of

recovering is better for both genders combined. Looking at the same data, we can read several apparently contradicting relations between probabilities.

Questions now arise: which relation should we trust? The "original" one, or the "reversed" one? How can we know for sure which one is original and which one is reversed? Which one accurately describes the effect of the new drug? These questions cannot be answered easily with classical statistics.

Next we show two examples with numbers based on the new drug scenario. The two examples are observationally completely equivalent, and yet they should be interpreted differently. The causal structure—if agreed upon—is the key to the correct interpretation and thus plays a crucial role in understanding the data. Without its help, it is not hard to tune a study in order to show that a certain wanted relation in the data holds.

### 2.1.2 Illustration of the Paradox

This subsection is inspired from Pearl (2000).

Suppose a new kind of drug is developed for a particular kind of illness, and a study is conducted to determine whether or not the new drug is effective. For 40 males and 40 females (which we distinguish with the variable $G$ for GENDER), we write down whether they were given the DRUG ($D$), and whether they eventually RECOVERED ($R$). Table 2.1 shows the sum of the results for both males and females, and Table 2.2 shows the results for males and for females separately.

| Both genders | Recovered ($R$) | Not recovered ($\neg R$) | Sum | Recovery rate |
|---|---|---|---|---|
| No drug ($\neg D$) | 16 | 24 | 40 | 40% |
| Drug ($D$) | 20 | 20 | 40 | 50% |
| Sum | 36 | 44 | 80 | |

TABLE 2.1: Results of the example study combined for males and females

| Females ($G = f$) | Recovered ($R$) | Not recovered ($\neg R$) | Sum | Recovery rate |
|---|---|---|---|---|
| No drug ($\neg D$) | 9 | 21 | 30 | 30% |
| Drug ($D$) | 2 | 8 | 10 | 20% |
| Sum | 11 | 29 | 40 | |
| **Males ($G = m$)** | **Recovered ($R$)** | **Not recovered ($\neg R$)** | **Sum** | **Recovery rate** |
| No drug ($\neg D$) | 7 | 3 | 10 | 70% |
| Drug ($D$) | 18 | 12 | 30 | 60% |
| Sum | 25 | 15 | 40 | |

TABLE 2.2: Results of the example study for males and females separately

Let us try to interpret these tables by answering the question, "is the new drug effective?" From the combined table, we read that taking the drug makes the recovery rate go up from 40% to 50%. If we wanted to sell the drug, we could stop here and answer the motivating question by saying "yes."

But looking at the females, we see that taking the drug actually *lowers* their recovery rate from 30% to 20%. We would then naturally expect the males to compensate for it and to recover drastically better when given the drug, which would be an intuitive explanation for the beneficial effect read from the combined table. Surprisingly, the drug also has a negative effect on the males, lowering their recovery rate from 70% to 60%.

There is something wrong in this study, although we chose 40 males and 40 females and gave the drug to 40 people with a control group of 40 other people. Formally, we have

$$p(R \mid D) > p(R \mid \neg D), \tag{2.1}$$

namely, the probability of recovering is higher if we take the drug. But conditioning on the gender reverses this relation:

$$\begin{aligned} p(R \mid D,\ G = f) &< p(R \mid \neg D,\ G = f) \\ p(R \mid D,\ G = m) &< p(R \mid \neg D,\ G = m), \end{aligned} \tag{2.2}$$

namely, if we know the gender, the probability of recovering is higher if we do not take the drug. Which shall we take as the correct characterization of the effect of the new drug?

Absurdly, if we use these probabilities to decide whether or not to give the drug to a new patient, we should give them the drug when their gender is unknown (as prescribed by Relation 2.1), but not give it to them as soon as we know their gender and can condition on it (as shown by Relations 2.2). If we use these results as a tool to evaluate the effect of our intervention of giving the drug, we are indeed in a paradoxical situation.

Looking more closely at the data, we see that the males recover naturally a lot better than the females. The study was biased by giving the drug to more males than females, making the good recovery rate of the males more important when given the drug, and the bad recovery rate of the females less important in the final sum. Said differently, the gender was used as a criterion for giving the drug or not, so that the natural better recovery rate of males is used to make the use of the drug appear beneficial in the combined table.

Once we have identified this fact and understood these (possibly) hidden relations between the variables, we feel that we *have* to condition on the gender to read the true effect of the drug, and that this drug is not beneficial for either a male or a female. The conditioning, by holding constant the value of the variable we condition on, "suppresses" its effect by holding it constant, too.

But let us now conduct a very similar study, and replace the variable GENDER by BLOOD PRESSURE ($B$). We know that the recovery also depends on the blood pressure, and that people with a high blood pressure ($B = h$) recover less than people with a normal blood pressure ($B = n$). Moreover, we know that the new drug influences blood pressure. Now suppose we obtain the same results as in Tables 2.2 and 2.1, GENDER being substituted by BLOOD PRESSURE. Is the conclusion of our study the same? Must we condition on $B$, just as we determined that we conditioned on $G$?

Our interpretation, this time, is that we must *not* condition on $B$. Conditioning holds $B$ constant: we thus remove the effect of the blood pressure on the recovery rate. But if we know that the drug does influence the blood pressure, we will want to include its influence on the recovery rate in our assessment of whether or not the drug is effective: leaving it out would be omitting all indirect influence "mediated" by our measurement of $B$ and would not accurately describe the drug effect.

Statistically, the two studies are the same. Only causal considerations differentiate them. In the next subsection, where we explicitly show the causal relationships between the variables, this discussion will look natural.

### 2.1.3 Causal Graphs

We first need some preliminary, intuitive but yet unmotivated notation to depict our causal assertions. The reader will find a formal definition and a discussion of the adequacy and of the properties of this notation in Section 3.1.

From now on, we will note $\mathbf{V}$ the set of all our observed variables. $X$, $Y$, $Z$, $W$, etc. are individual variables taken from $\mathbf{V}$.

**Definition 2.1.1 (Causal graph (preliminary))** *A* causal graph *is a directed acyclic graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$*, where the nodes* $\mathbf{V}$ *are the variables of interest in a particular study, and where each directed arc* $(X \to Y) \in \mathbf{E}$ *with* $X, Y \in \mathbf{V}$ *is interpreted as "the variable* $X$ *has a causal effect on the variable* $Y$ *which is unmediated by the other variables."*

In a causal graph, we thus represent direct cause–effect relationships with arrows between the variables. Note that we define the graph to be acyclic, i.e., we assume that our system does not contain causal feedback loops. Causal graphs containing cycles vastly remain an open area for research.

Using this definition, Figure 2.1 shows the causal links in our first study in Subsection 2.1.2, and Figure 2.2 is the causal graph for the study where GENDER was replaced by BLOOD PRESSURE.

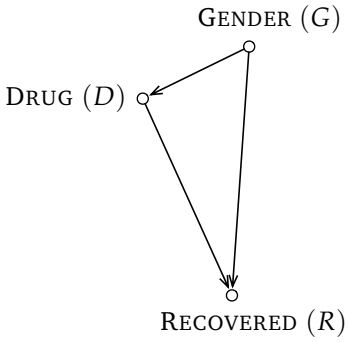



FIG. 2.1: Causal graph with node GENDER

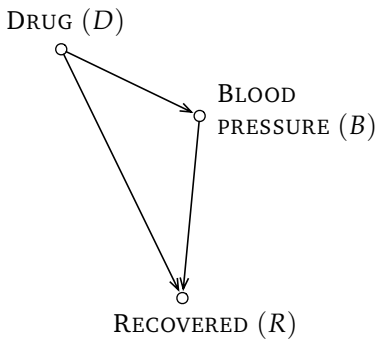



FIG. 2.2: Causal graph with node BLOOD PRESSURE

Figure 2.1 shows an arc from $G$ to $D$ because we determined that in the study, the drug was given to more males than females, and thus that $G$ was used to determine $D$.

Now, the total causal effect of a variable $X$ on another variable $Y$ is, intuitively, the sum over the effect of every directed path between $X$ and $Y$. In Figure 2.1, this only includes the arc $D \to R$. In Figure 2.2, we must add the chain $D \to B \to R$.

In this simple example, to answer the question "to condition or not to condition" on a variable $Z$, we can simply look whether or not $Z$ lies on a causal path between $D$ and $R$. If it is the case, then we do not condition. If it is not the case, we can do it—and *must* do it whenever $Z$ is a *confounding factor* for $D$ and $R$, e.g., a common cause, influencing both. The justification for that can be found in the discussion in Subsection 2.1.2: if we did not condition on the gender $G$ in Figure 2.1, we would silently include arc $G \to R$ in our assessment of the effect of $D$ on $R$, which is not correct.

### 2.1.4 Assessing Causal Effect

The previous illustration of Simpson's paradox is a striking example showing that observation must not be mistaken for intervention: we have seen that assessing the effect of variable $X$ on variable $Y$ cannot be readily expressed with a standard probability conditioning such as $p(Y \mid X)$. Depending on the causal context, it may well be some other form of conditioning. In Pearl (2000), a new notation, called *"do" notation*, and a corresponding new type of probability conditioning, the *"do" conditioning*, are used. The following notation:

$$p(Y = y \mid do(X = x)) \tag{2.3}$$

(or the abbreviated form $p(y \mid do(x))$) represents the probability that $Y = y$ given that we force variable $X$ to take value $x$.[1] The three rules of the *do* calculus allow to find a closed-form expression for a *do* expression given a causal structure whenever possible. Depending on the causal structure, some *do* expressions cannot be evaluated, while in some other cases, it is possible to assess causal effects involving unobserved variables (but nevertheless modeled and present in the causal graph).

Using the rules of the *do* calculus, we find, indeed, that in our first example, the total effect of the drug on the recovery rate must be evaluated with the help of $p(R \mid do(D)) = p(R \mid D,\ G)$, whereas in the second example we have $p(R \mid do(D)) = p(R \mid D)$.

It is possible to find a systematic way to expand a *do* expression; moreover, it was recently shown (Valorta & Huang, 2006b) that the rules are complete, i.e., that there exists a sequence of transformations leading to a closed-form expression involving only observable quantities if and only if the causal effect is identifiable. Whether it is identifiable can also be tested systematically with graphical conditions (Valorta & Huang, 2006a).

The detailed rules of the *do* calculus can be found in Section 3.3, along with example. We do not extensively discuss it as our focus is causal structure construction, but it is worth mentioning that it is possible to algebraically reason on causal graphs in order to assess causal effects—this is indeed the final goal beyond structure construction. The interested reader is invited to read Chapter 3 of Pearl (2000) and look at the references listed in it or at the numerous papers based on it.

## 2.2 CAUSATION AND ITS ASYMMETRY: CHALLENGES

The missing step is now to go from observational data to a causal network, i.e., to find tools, rules and algorithms to construct the graph structure. Ideally, for each variable, we would like to know the set of its direct causes (graphically, its parents) and the set of its direct effects (its children).

In the literature however, there exists no "test of causality."[2] What does exist are similarity measures between two variables—like Pearson's correlation, or Shannon's mutual information—or between more than two variables, like multiple correlation. The problem is that many of these

[1] Other notations in the literature include $p(Y = y \mid set(X = x))$, $p(Y = y \mid X = \hat{x})$, $p(Y = y \parallel X = x)$ or $p(Y = y \mid X \leftarrow x)$.

[2] A concept called *Granger causality* exists (Granger & Engle, 1987), which is a technique for determining whether a given time series helps predicting the value of another with a certain time lag. We then say that $X$ Granger-causes $Y$. We find that this concept has more to do with feature selection or with forecasting techniques than with causation. Granger himself (Granger, 1988) is very careful to delimit his notion of causality, in order to distinguish it from controllability.

similarity measures are symmetrical—or if they are not, their lack of symmetry cannot be interpreted causally. Causation, on the other side, is intrinsically asymmetrical, with causes and effects (again, we assume no causal feedback loops in our system).

In order to temporarily simplify our task, we will first construct an undirected graph. In an undirected graph, causes and effects are not differentiated; rather, we have for each node a set of neighbors, which can be either causes and effects. Every relation is now symmetrical. We show with a simple example that even in this case, symmetrical similarity measures fall short of being able to detect such a structure.

Consider the following situation: whenever it rains and I have to go out, I will take my umbrella with me. If I travel with my umbrella, I might forget it in the bus or in the office. We can represent this with the simple causal graph in Figure 2.3, which we will try to recover.

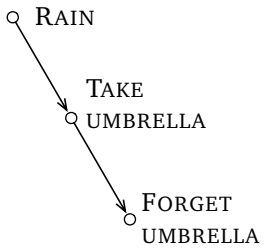



FIG. 2.3: An example of chained causation

As before, the arrows mean "has a causal influence on" (and not "implies" or "is the physical cause for"—by "causal", we mean, "causal with respect to the granularity level of our study").

Suppose each day, a data row is recorded with the (boolean) values of the three mentioned variables. Looking at all possible pairs with similarity measures, we find that every variable is positively correlated with every other. If we use this criterion to build the graph, we will end up with a fully connected graph, which is uninformative. We could try to take into account only the strongest correlations, and forget about the others (which would lead to the problem of determining an adequate threshold). But even then, there are situations where this approach is incorrect. Assuming that Figure 2.4 represents a true causal graph, we might well detect a stronger correlation between $X$ and $Y$ than between any other variable pair, including those linked by an arc. The contributions of the causal paths through $Z$, $U$ and $W$ can yield a high similarity between $X$ and $Y$, although no direct causation exists.

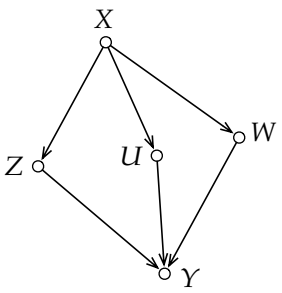



FIG. 2.4: A "diamond-shaped" causal structure

The simple similarity measure does not help much in this situation. Actually, it is even worse than

that: as far as the causal structure is concerned, a high similarity between two variables $X$ and $Y$ can mean any of the following:

1. $X$ causes $Y$ ($X \rightarrow Y$);
2. $Y$ causes $X$ ($Y \rightarrow X$);
3. $X$ indirectly causes $Y$ through some causal chain ($X \rightarrow Z_1 \rightarrow Z_2 \rightarrow \cdots \rightarrow Z_n \rightarrow Y$);
4. $Y$ indirectly causes $X$ through some causal chain ($Y \rightarrow Z_1 \rightarrow Z_2 \rightarrow \cdots \rightarrow Z_n \rightarrow X$);
5. There is a common cause $L$ for both $X$ and $Y$ ($X \leftarrow L \rightarrow Y$);
6. There is selection bias which makes $X$ and $Y$ appear correlated while they are not;
7. Any combination of the above.

If a causal chain links $X$ to $Y$, there is no major problem if we do not observe some variables in the chain (provided the unobserved variables do not interact with other parts of the system). The number of variables we include in our analysis determines the granularity of the analysis. Omitting them and redrawing the graph leads to similar conclusions on the structure of the data. But unobserved common causes are a problem, because they introduce confounding which can be mistaken for causation, and drastically change the way the effects of interventions are assessed. We will come back to unobserved common causes, called *latent variables*, as we discuss construction algorithms in Chapter 4.

Anyway, as we can see, there is little hope of constructing a causal network, even if undirected, with the only help of similarity measures. Let us sum up our main problems.

- **Problem 1:** How can we know if there is selection bias, and and how do we deal with it?
- **Problem 2:** How can we detect latent variables?
- **Problem 3:** How can we build the undirected causal structure?
- **Problem 4:** How can we direct the undirected causal structure?

In this study, we assume (1) that there is no selection bias, partially assume (2) that there are no latent variables, and show the best we can do today to solve (3) and (4). Some basic notions are now described in the next section.

## 2.3 BASICS OF STRUCTURE IDENTIFICATION

We now introduce the idea of holding a variable constant to compute what is called a *conditional similarity measure* and show how this is the key to structure construction.

### 2.3.1 Constructing the Undirected Structure

We start with Problem 3. In particular, how can we distinguish, looking at $X$ and $Y$, the case $X \rightarrow Y$, where we *must* add an arc between $X$ and $Y$, from the case $X \rightarrow Z \rightarrow Y$, where a causal chain mediated by $Z$ exists and thus where we must *not* include a direct arc from $X$ to $Y$?

The idea is to try to remove the effect of the variable we suspect might be mediating the causal link. If we observe, when $Z$ is fixed, that there still exists a strong association between $X$ and $Y$, then we know that there exists a link between $X$ and $Y$ that is not mediated by $Z$. It might be

a direct link, it might be a chained link—and it still might include a latent common cause (and, of course, we cannot know whether $X$ must be interpreted as a cause and $Y$ as an effect, or vice versa). If, on the contrary, the association between $X$ and $Y$ systematically vanishes when $Z$ is held constant at a certain value, then we can rule out the possibility of a direct link between $X$ and $Y$. Following this principle, it seems that it is easier to discard a link than confirm its existence.

Interestingly, this technique of holding the mediating variable constant to differentiate chained causation from direct causation can also be used to differentiate a common-cause case from direct causation, i.e., to know whether we have $X \leftarrow Z \rightarrow Y$ or $X \rightarrow Y$. In both cases, we will detect a significant association between $X$ and $Y$, but in the former case, holding $Z$ constant will make it vanish and will thus forbid a link between $X$ and $Y$.

It is a both a good thing and a bad thing that the same test should allow us to solve two different problems, namely the chained causation and the common cause. On the one hand, we need not perform two tests to know whether we can really rule out the direct link between $X$ and $Y$, but on the other hand, if we do rule it out, we do not know how to orient the arrows around $Z$, because the two cases are indistinguishable. This is part of what is called the *causal underidentification* problem. We will mention it again in this chapter and throughout Chapter 3.

There *is* a special case however, called *V-structure*, which gives us a hint about which variables are causes and which are effects.

### 2.3.2 V-Structures & Explaining Away: Directing Links

We now discuss Problem 4.

**Definition 2.3.1 (V-structure)** *Whenever two nonadjacent variables $X$ and $Y$ are both causes for a third variable $Z$, we say that the tuple $(X, Z, Y)$ forms a* V-structure *in the corresponding causal graph. $Z$ is then called a* collider *for the pair $(X, Y)$.*

Let us turn to an example to get a better intuition of the particular case of V-structures. Suppose we have the causal graph shown in Figure 2.5.

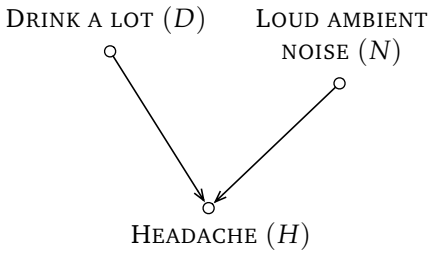



FIG. 2.5: A causal graph containing a V-structure

In this scenario, a headache is causally influenced by whether a lot of alcohol was drunk and/or by a loud ambient noise. We further suppose that $D$ is not associated with $N$, i.e., that drinking a lot is neither a cause nor an effect of the loud ambient noise (we do not discuss the validity of this assumption).

This case is quite different from a chained causation and from a common cause, because we now have the property that the two variables at the extremes of the undirected structure are *not* associated with each other.

But the most interesting phenomenon is that they become correlated if their common effect $H$ is held constant. If I have a headache and I have not been drinking, then it was most probably caused by the loud ambient noise. Conversely, suppose I have a headache and there is a loud ambient noise: knowing this second fact $N$ makes the event of having drunk a lot $D$ less likely. We say that $D$ is *explained away* by $N$. Knowing the value of the collider $H$ creates a dependency between its parents $D$ and $N$.

Creating the association by holding a variable constant happens in the V-structure case, but also happens with descendants of colliders. Intuitively, this seems logical; if, due to a rough granularity level, we miss the true common cause and only observe one of its direct effects, which is strongly associated with it, we are in the same case as the one discussed with chained causation. Leaving out a mediating variable in the middle of the directed chain does not impair the conclusions drawn from the causal graph.

The detection of V-structures is, statistically, the only way to differentiate causes from effects, and is thus a crucial point. It is often the case that additional edges can be directed after the V-structure detection phase, using constraints such as graph acyclicity. This step is detailed later when reviewing causal construction algorithms in Section 4.4.

The last point that should be addressed before formalizing the framework is how to compute a conditional similarity measure. For simplicity and historical reasons related to Bayesian networks (see Subsection 3.2.1), we now present statistical conditional independence. This ternary relation also has a set of properties useful for discussing the adequacy of graphs in Section 3.1.

### 2.3.3 Conditional Independence

To determine the degree of association between two variables when holding a third variable (or a set of other variables) constant, we now examine well-studied tools from probability theory. In our efforts to build the network structure, we need a similarity measure and conditional similarity measure: independence and conditional independence are key concepts.

**Definition 2.3.2 (Independence)** *Two random variables $X$ and $Y$ are said to be* independent *with respect to a probability distribution $p(\cdot)$, noted $(X \perp\!\!\!\perp Y)_p$, if and only if, for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$:*

$$p(X = x \mid Y = y) = p(X = x). \tag{2.4}$$

**Definition 2.3.3 (Conditional Independence)** *Two random variables $X$ and $Y$ are said to be* conditionally independent given $\mathbf{Z}$ *with respect to a probability distribution $p(\cdot)$, noted $(X \perp\!\!\!\perp Y \mid \mathbf{Z})_p$, if and only if, for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $\mathbf{z} \in \boldsymbol{\mathcal{Z}}$:*

$$p(X = x \mid Y = y,\ \mathbf{Z} = \mathbf{z}) = p(X = x \mid \mathbf{Z} = \mathbf{z}). \tag{2.5}$$

When the context is clear, the subscript $_p$ is dropped, so that we write $(X \perp\!\!\!\perp Y)$ and $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$, respectively.

This can be interpreted as the irrelevance of $Y$ about $X$ once we know $\mathbf{Z}$: knowing $Y = y$ does not reduce the uncertainty on $X$ if it is given that $\mathbf{Z} = \mathbf{z}$. If $\mathbf{Z} = \emptyset$, then $X$ and $Y$ are *unconditionally independent* or simply *independent* according to the previous definition.

Conditional independence is then a ternary relation which has the following properties.[3] Here $X$, $Y$, $Z$ and $W$ can also be sets of variables (boldface is left out for readability reasons). Considering the special case $Z = \emptyset$ helps understanding these properties.

- **Symmetry**

$$(X \perp\!\!\!\perp Y \mid Z) \iff (Y \perp\!\!\!\perp X \mid Z) \tag{2.6}$$

  If, given $Z$, we do not learn anything on $X$ knowing $Y$, we will not learn anything about $Y$ knowing $X$ either.

- **Decomposition**

$$(X \perp\!\!\!\perp W \cup Y \mid Z) \implies (X \perp\!\!\!\perp W \mid Z) \text{ and } (X \perp\!\!\!\perp Y \mid Z) \tag{2.7}$$

  Special case with $Z = \emptyset$: $(X \perp\!\!\!\perp W \cup Y) \Rightarrow (X \perp\!\!\!\perp Y)$. If the union of $W$ and $Y$ is irrelevant to $X$ (knowing $Z$), then clearly, knowing only $Y$ (or only $W$) will not be relevant to $X$ either.

- **Weak Union**

$$(X \perp\!\!\!\perp W \cup Y \mid Z) \implies (X \perp\!\!\!\perp W \mid Y \cup Z) \text{ and } (X \perp\!\!\!\perp Y \mid W \cup Z) \tag{2.8}$$

  Special case with $Z = \emptyset$: $(X \perp\!\!\!\perp W \cup Y) \Rightarrow (X \perp\!\!\!\perp W \mid Y)$. If the union of $W$ and $Y$ is irrelevant to $X$ (knowing $Z$), then conditioning on either one cannot help the other become relevant.

- **Contraction**

$$(X \perp\!\!\!\perp W \mid Y \cup Z) \text{ and } (X \perp\!\!\!\perp Y \mid Z) \implies (X \perp\!\!\!\perp W \cup Y \mid Z) \tag{2.9}$$

  Special case with $Z = \emptyset$: $(X \perp\!\!\!\perp W \mid Y)$ and $(X \perp\!\!\!\perp Y) \Rightarrow (X \perp\!\!\!\perp W \cup Y)$. If $W$ is judged irrelevant to $X$ after learning $Y$, which we know is independent from $X$, then $W$ is also unconditionally irrelevant.

- **Intersection**

$$(X \perp\!\!\!\perp W \mid Y \cup Z) \text{ and } (X \perp\!\!\!\perp Y \mid W \cup Z) \implies (X \perp\!\!\!\perp W \cup Y \mid Z) \tag{2.10}$$

  This only holds if the joint probability distribution $p(\mathbf{V})$ is strictly positive. Special case with $Z = \emptyset$: $(X \perp\!\!\!\perp W \mid Y)$ and $(X \perp\!\!\!\perp Y \mid W) \Rightarrow (X \perp\!\!\!\perp W \cup Y)$. If both $Y$ and $W$ are irrelevant to $X$ given one another, then neither the one nor the other (nor their union) will be relevant to $X$.

### 2.3.4 Linking Causation to Conditional Independence

We link causation to conditional independence by deriving implications and equivalences between them. The following statements are also used to design graph construction algorithms and to prove that they construct a correct graph.

[3] These properties are also known as the *graphoid axioms* (Galles & Pearl, 1997).

If we know that in a certain system, variable $X$ has a direct causal influence on $Y$, noted $X \to Y$, we can say the following in terms of conditional independence, with $X, Y \in \mathbf{V}$:

$$X \to Y \implies \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big). \tag{2.11}$$

This means that when there is a direct causation between $X$ and $Y$, it is impossible to find a subset of variables that would make $X$ and $Y$ independent. The converse of this implication, which would be an extremely useful formulation for causation detection, does not hold. We have instead, with $X, Y, Z \in \mathbf{V}$:

$$\big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big) \implies X \to Y \text{ or } Y \to X \text{ or } X \leftarrow L \to Y. \tag{2.12}$$

This highlights again causal underidentification, in that the same conditional independence condition has several causal interpretations. Here, we reserve the possibility of a hidden common cause in order to explain a given dependency. Certain assumptions (see Section 4.3) allow us to rule out this possibility and simplify our task.

A V-structure—two causes $X$, $Y$ with the same effect $Z$—implies:

$$\begin{aligned} & X \to Z \leftarrow Y \\ \implies & \big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big). \end{aligned} \tag{2.13}$$

The exact converse does not hold either, but instead, we have:

$$\begin{aligned} & \big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : \overbrace{(X \perp\!\!\!\perp Y \mid \mathbf{S})}^{\text{Condition 1}} \text{ and } \overbrace{(X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})}^{\text{Condition 2}}\big) \\ \implies & X \leftarrow\!\cdot\!\rightarrow Z \leftarrow\!\cdot\!\rightarrow Y \text{ or } Z \text{ is an effect of some } W \text{ such that } X \leftarrow\!\cdot\!\rightarrow W \leftarrow\!\cdot\!\rightarrow Y. \end{aligned} \tag{2.14}$$

The notation $X \leftarrow\!\cdot\!\rightarrow Z$ means "any causal chain between $X$ and $W$ that does not contain a V-structure." $X \to Z \leftarrow Y$ is a special case of $X \leftarrow\!\cdot\!\rightarrow Z \leftarrow\!\cdot\!\rightarrow Y$.

Condition 1 implies that $X$ and $Y$ must be nonadjacent, otherwise Implication 2.11 states that no set $\mathbf{S}$ can be found such that $X$ and $Y$ become independent. Condition 2 ensures that a dependency is created by including the additional variable in the conditioning set. In theory, it is not a problem that Implication 2.14 does not only identify colliders but also their descendents. We might be afraid of creating V-structures where there are none, but we can actually check before creating them that the undirected edges do exist using Implication 2.11. Thus, if we assume that there are no latent variables, there is a necessary and sufficient condition for the existence of a V-structure $X \to Z \leftarrow Y$:

$$\begin{aligned} & X \to Z \leftarrow Y \\ \iff & \Big(\big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big) \\ & \quad \text{and } \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Z\} : (X \not\perp\!\!\!\perp Z \mid \mathbf{S})\big) \\ & \quad \text{and } \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{Y, Z\} : (Y \not\perp\!\!\!\perp Z \mid \mathbf{S})\big)\Big). \end{aligned} \tag{2.15}$$

If we drop the assumption that there are no latent variables, we must replace the first line of

Equivalence 2.15 by

$$X \to Z \leftarrow Y \text{ or } X \leftarrow L \to Z \leftarrow Y \text{ or } X \to Z \leftarrow L \to Y \text{ or } X \leftarrow L_1 \to Z \leftarrow L_2 \to Y,$$

where in all cases, we find converging arrows around $Z$, but where causation between $X$ and $Z$ and between $Y$ and $Z$ can be mediated by unobserved common causes.

Implications 2.12 and 2.14 are in many cases not enough to obtain a fully directed graph. This underidentification of the causal structure, already mentioned in Subsection 2.3.1, leads to the definition in Subsection 3.1.4 of equivalence classes of graphs, i.e., of families of graphs, all members of which are compatible with a given set of conditional independencies.

**Summary**

*We motivated the need for causal structure by showing that Simpson's paradox can be solved with causal considerations, and in particular with a causal graph, representing direct cause–effect relationships between the variables of interest. We mentioned that there exists a calculus of interventions called* do *calculus to reason algebraically over causal graphs. We then turned to the problem of causal graph construction and highlighted several challenges in this task, discussing selection bias, confounding factors and causal underidentification. We identified the V-structure pattern as the only case where causes can be formally differentiated from effects. We finally introduced conditional independence as our main tool for constructing the network and established properties connecting causal structure to conditional independence.*

CHAPTER

3

# Causality Formalized

*This chapter formalizes the structure identification problem. We first look at graphical models to represent sets of conditional independence relations and explain their respective limitations. We motivate why directed acyclic graphs are the preferred way to talk about causality and what assumptions allow us to attach a causal interpretation to them. Interventions can also be directly represented on graphs. We finally make the link with Bayesian networks and structural equation models, and show how interventions can be modeled with them.*

## 3.1 Graphs and Causation

Following the implications stated in Subsection 2.3.4, we will use conditional independence tests to construct the structure of the graphs. For now, we have used directed graphs to represent our simple causal models for discussion. Next, we show that they are a sound choice with respect to the properties of conditional independence.

Conditional independence is a ternary relation, which has properties described in Subsection 2.3.3. If we base graph construction on this relation, we have to find some ternary relation related to our graphical representation which has properties as similar as possible to conditional independence. Ideally, we should find an isomorphism mapping conditional independence to some graphical criterion. If no isomorphism can be found, then there are cases where not all conditional independencies or dependencies can be represented by the graphical model. These cases have to be identified and discussed.

In this section, we examine both undirected graphs and directed acyclic graphs as graphical methods to represent independencies, and discuss their adequacy.

**Graphs & Notation** We quickly review usual notations and definitions pertaining to graphs. A *graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$ consists of the vertices $\mathbf{V}$ and the edges $\mathbf{E}$. We adopt the convention that for an undirected edge $X - Y$ we have $(X, Y), (Y, X) \in \mathbf{E}$, and that a directed arc

$X \to Y$ is characterized by $(X,Y) \in \mathbf{E}, (Y,X) \notin \mathbf{E}$. This definition of a graph can be extended to accommodate bidirected arcs or marked arcs, as required by various algorithms. In general, we write *edge* for an undirected link and *arc* for a directed link—*link* referring to either one, depending on the context.

In a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, we define, for each node $X \in \mathbf{V}$:

- $\mathbf{Pa}_{\mathcal{G}}(X) = \{Y \mid (Y,X) \in \mathbf{E}\}$ as the *parents* of $X$;
- $\mathbf{Ch}_{\mathcal{G}}(X) = \{Y \mid (X,Y) \in \mathbf{E}\}$ as the *children* of $X$;
- $\mathbf{Bd}_{\mathcal{G}}(X) = \mathbf{Pa}_{\mathcal{G}}(X) \cup \mathbf{Ch}_{\mathcal{G}}(X)$ as the *boundary* of $X$, i.e., direct neighbors of $X$;
- $\mathbf{An}_{\mathcal{G}}(X) = \mathbf{Pa}_{\mathcal{G}}(X) \cup \mathbf{An}_{\mathcal{G}}(\mathbf{Pa}_{\mathcal{G}}(X))$, recursively defined as the *ancestors* of $X$;
- $\mathbf{De}_{\mathcal{G}}(X) = \mathbf{Ch}_{\mathcal{G}}(X) \cup \mathbf{De}_{\mathcal{G}}(\mathbf{Ch}_{\mathcal{G}}(X))$, recursively defined as the *descendants* of $X$;
- $\mathbf{Nd}_{\mathcal{G}}(X) = \mathbf{V} \setminus \mathbf{De}_{\mathcal{G}}(X) \setminus \mathbf{Pa}_{\mathcal{G}}(X) \setminus \{X\}$ as the *nondescendants* of $X$;
- $\mathbf{Mb}_{\mathcal{G}}(X) = \big(\mathbf{Pa}_{\mathcal{G}}(X) \cup \mathbf{Ch}_{\mathcal{G}}(X) \cup \mathbf{Pa}_{\mathcal{G}}(\mathbf{Ch}_{\mathcal{G}}(X))\big) \setminus \{X\}$ as the *Markov blanket* of $X$.

When the context is clear, we drop the subscript $_{\mathcal{G}}$, as well as when noting a specific instantiation of the variables contained in one of these sets, which we note, e.g., $\mathbf{pa}_X$ or $\mathbf{ch}_X$.

### 3.1.1 Undirected Graphs

We saw that conditional independence is symmetric. Correlation and mutual information, which are important similarity measures, are also symmetrical. Why not then use undirected graphs, in which there is no asymmetry due to the presence of directed arcs? Actually, this has been looked into, and can be found in the literature under the name *Markov random fields* or *Markov networks*.

Relevant to our problem is the separation criterion. Intuitively, we want two unconditionally dependent variables $X$ and $Y$ to be linked, while having the set of variables making two other (otherwise dependent) variables $Z$ and $W$ independent coincide with the graphically separating set of nodes.

**Definition 3.1.1 (Separation)** *Two nodes $X$, $Y$ are* separated *by $\mathbf{Z}$ ($X, Y \notin \mathbf{Z}$) in an undirected graph $\mathcal{G}$, written $\mathrm{Sep}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$, if and only if every path from $X$ to $Y$ contains at least one node $Z_i \in \mathbf{Z}$. If $X$ and $Y$ are not separated by $\mathbf{Z}$, they are* connected*: $\mathrm{Conn}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$.*

Before discussing the adequacy of this ternary relation as graphical representation of conditional independence, we define a set of properties to help assess whether a graph proves to represent the set of all conditional independencies identified in a joint probability distribution $p(\cdot)$. The three *Markov properties* listed below describe what implications hold if the respective property is verified for a graph $\mathcal{G}$ and the corresponding distribution $p(\mathbf{V})$. They are important because they also determine what kind of sets of conditional independences can be represented with undirected graphs.

- **Undirected Pairwise Markov Property (UP)**

$$\forall X, Y \in \mathbf{V} : \big(\mathrm{Sep}(X, Y \mid \mathbf{V} \setminus \{X,Y\})_{\mathcal{G}} \implies (X \perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X,Y\})_p\big) \tag{3.1}$$

  $\mathrm{Sep}(X, Y \mid \mathbf{V} \setminus \{X,Y\})$ denotes two nodes $X$, $Y$ that are not directly connected. If (UP) holds, then two nonadjacent nodes represent variables that are independent given all other variables.

- **Undirected Local Markov Property (UL)**

$$\forall X \in \mathbf{V} : (X \perp\!\!\!\perp \mathbf{V} \setminus \mathbf{Bd}_{\mathcal{G}}(X) \setminus \{X\} \mid \mathbf{Bd}_{\mathcal{G}}(X))_p \tag{3.2}$$

  The set $\mathbf{Bd}(X)$ denotes the *boundary* of the node $X$, which is the set of all direct neighbors of $X$. In terms of directed graphs, $\mathbf{Bd}(X)$ corresponds to the *Markov blanket* of $X$ (see Subsection 3.1.2). If (UL) holds, then a given variable $X$ is independent of all others given its direct neighbors, i.e., there are no hidden dependencies influencing it that do not appear in the graph.

- **Undirected Global Markov Property (UG)**

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V} : \big(\mathrm{Sep}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_{\mathcal{G}} \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_p\big) \tag{3.3}$$

  If (UG) holds, then whenever $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in the graph, we have a conditional independency of $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ in the distribution. This means that all independencies represented by the graph are true, and, by contraposition, that all dependencies in the distribution are represented by the graph. It does not mean, however, that the graph models *all* independencies that are present in the distribution; i.e., there may be extra dependencies in the graph that are not present in the distribution. (Below, we explain the concept of *I-map*: (UG) means that the graph is an I-map of the distribution.)

The following holds: (UG) $\Rightarrow$ (UL). With positive distributions, (UP) $\Rightarrow$ (UL) $\Rightarrow$ (UG). This means that with positive distributions, we can use the pairwise Markov property to construct a graph complying with the global Markov property.

To further be able to characterize how well a graph fits a set of conditional independencies found in a dataset, we define I-maps, D-maps and P-maps.

**Definition 3.1.2 (Undirected I-map)** *An undirected graph $\mathcal{G}$ is called an* undirected independence map *or* undirected I-map *of a joint probability distribution $p(\cdot)$ if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \big(\mathrm{Sep}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \implies (X \perp\!\!\!\perp Y \mid \mathbf{Z})_p\big) \tag{3.4}$$

*(and, by contraposition, $(X \not\perp\!\!\!\perp Y \mid \mathbf{Z})_p \Rightarrow \mathrm{Conn}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$). A graph is an I-map of a distribution if and only if the global Markov property is satisfied.*

In words, all independencies represented in an I-map must hold in the probability distribution, and all dependencies in the distribution are shown in the I-map. A complete graph is trivially always an I-map because it does not depict any independencies and thus represents (probably more than) all dependencies of the distribution. A *minimal undirected I-map* is an I-map such that removing any edge from it would make it lose the I-map property; in other words, in order to make an I-map minimal, we keep removing the extra dependencies that appear in the graph and not in the distribution.

**Definition 3.1.3 (Undirected D-map)** *An undirected graph $\mathcal{G}$ is called an* undirected dependence map *or* undirected D-map *of a joint probability distribution $p(\cdot)$ if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \big(\mathrm{Conn}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \implies (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})_p\big) \tag{3.5}$$

*(and, by contraposition, $(X \perp\!\!\!\perp Y \mid \mathbf{Z})_p \Rightarrow \mathrm{Sep}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$).*

In words, all dependencies represented in an D-map must hold in the probability distribution, and all independencies in the distribution are shown in the D-map. A graph without any edges is a trivial D-map because it does not depict any dependencies and thus represents all independencies of the distribution. A *maximal undirected D-map* is an D-map such that adding any extra edge to it would make it lose the D-map property.

**Definition 3.1.4 (Undirected P-map)** *An undirected graph $\mathcal{G}$ is a* undirected perfect map *or* undirected P-map *of a joint probability distribution $p(\cdot)$ when it is both an undirected I-map and an undirected D-map, i.e., if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \left( \text{Sep}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \iff (X \perp\!\!\!\perp Y \mid \mathbf{Z})_p \right) \tag{3.6}$$

*(and thus* $\text{Conn}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \Leftrightarrow (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})_p$*).*

Our goal is to build a P-map of the probability distribution we have to explore, such that it best represents the distribution. Is it always possible, given that any positive distribution meets the independence properties described above? No, it is not. In particular, the joint probability distribution of the data would have to satisfy the *strong union* property: $(X \perp\!\!\!\perp Y \mid \mathbf{Z}) \Rightarrow (X \perp\!\!\!\perp Y \mid \mathbf{Z} \cup \{W\})$. This is violated by models where two causes have a single effect—the V-structure pattern. A special case is that two variables unconditionally independent must always stay independent, which is not generally true. Any *induced dependency* which appears when information about an additional variable becomes available cannot be represented by an undirected P-map. Distributions that can be represented by an undirected P-map are called *undirected graph-isomorphic*. The corresponding P-map is then unique.

We see that undirected graphs are not well-suited for our causality representation problem.

### 3.1.2 Directed Acyclic Graphs

Let us now observe the expressiveness of directed acyclic graphs (DAGs). First, if we want to avoid falling into the same limitation as for the undirected graphs, we have to elaborate on the concept of separation and refine it.

**Definition 3.1.5 (*d*-Separation)** *Two nodes $X$, $Y$ are* $d$-separated *by $\mathbf{Z}$ ($X, Y \notin \mathbf{Z}$) in a DAG $\mathcal{G}$,*[1] *written* $\text{dSep}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$*, if every path from $X$ to $Y$ is blocked by $\mathbf{Z}$. A path is* blocked *if at least one of the nodes on the path is blocked by $\mathbf{Z}$. Whether a node is blocked or not by $\mathbf{Z}$ depends on the direction of the arcs to and from the node:*

1. *If the node (with respect to the considered path) is diverging or serially connected, then the node is blocked if it is in $\mathbf{Z}$;*
2. *If, on the contrary, the node is converging (both arcs are oriented toward it), it is blocked if neither itself nor one of its descendants is in $\mathbf{Z}$.*

*If $X$ and $Y$ are not $d$-separated by $\mathbf{Z}$, they are* $d$-connected*:* $\text{dConn}(X, Y \mid \mathbf{Z})_{\mathcal{G}}$.

To determine whether or not $\text{dSep}(X, Y \mid \mathbf{Z})$ holds, we can either find an unblocked path from $X$ to $Y$ to prove that they are $d$-connected, or show that all possible paths from $X$ to $Y$ are blocked to

[1] The "d" stands for "dependence."

prove that they are $d$-separated. The reader is invited to test their understanding of this criterion by verifying in Figure 3.1 that $\mathrm{dSep}(X, Y \mid \mathbf{Z})$, $\mathbf{Z}$ being the set of the nodes marked with bullets (•), holds in case (c) only.

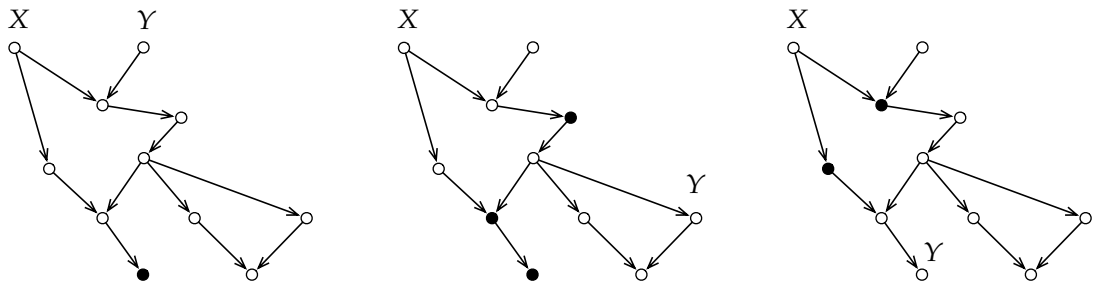



FIG. 3.1: Test cases for the $d$-separation criterion

In order for the $d$-separation criterion to make sense, interpret $d$-connectedness as variable dependence. The first condition stated in the definition is obvious and corresponds to the separation criterion in undirected graphs: whenever we know the value of a variable in a dependence chain, variables before and after it lose their dependencies (serially connected case); the same holds if two otherwise independent variables $X$, $Y$ are dependent on the same variable $Z$. Once we know $Z$, the dependency between $X$ and $Y$ is lost (diverging case).

The converging case is more special since it is the information about a certain variable that makes two others dependent, and is exactly the case described when discussing the limitations of the undirected graph representation. Suppose that two independent variables $X$, $Y$ are given, with a third variable $Z$ being dependent on $X$ and $Y$, such that we have $X \rightarrow Z \leftarrow Y$. Then we have $\mathrm{dConn}(X, Y \mid Z)$ because of the explaining away phenomenon (see Subsection 2.3.2).

As with undirected graphs, we can discuss the Markov properties for DAGs.

- **Directed Pairwise Markov Property (DP)**

$$\forall X \in \mathbf{V}, \forall Y \in \mathbf{Nd}_{\mathcal{G}}(X) : (X \perp\!\!\!\perp Y \mid \mathbf{Nd}_{\mathcal{G}}(X) \setminus \{Y\})_p \tag{3.7}$$

  $X$ and $Y$ are *nonadjacent* nodes. The set $\mathbf{Nd}(X)$ represents the *nondescendents* of $X$ in the graph. If (DP) holds, then all variables are independent of any other ancestor that is not a direct parent given all its nondescendents. Note that $Y$ must now be part of $\mathbf{Nd}(X)$, which is less restrictive than the corresponding undirected pairwise Markov property.

- **Directed Local Markov Property (DL)**

$$\forall X \in \mathbf{V} : (X \perp\!\!\!\perp \mathbf{Nd}_{\mathcal{G}}(X) \mid \mathbf{Pa}_{\mathcal{G}}(X))_p \tag{3.8}$$

  If (DL) holds, then any variable, given its parents, is independent of all its nondescendents. Note that, here again, we only define these independence relations for a variable relative to its nondescendents. (DL) means that in the set of nondescendents, once the direct parents are given, there is no hidden influence from any other variable. This is equivalent to stating that each variable $X$ is conditionally independent of all others given its *Markov blanket*

$\mathbf{Mb}(X)$, which is the set of its parents, children and children's parents: $(X \perp\!\!\!\perp Y \mid \mathbf{Mb}(X))$ for all $Y \notin \mathbf{Mb}(X)$. We need to include the children's parents (or *spouses* of $X$) in order to prevent them from explaining $X$ away.

- **Directed Global Markov Property (DG)**

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V} : \big( \mathrm{dSep}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) \big) \tag{3.9}$$

(DG) is identical to (UG) if we substitute separation by $d$-separation.

We have (DG) $\Leftrightarrow$ (DL) $\Rightarrow$ (DP), and (DP) $\Rightarrow$ (DL) if the distribution is strictly positive. Note that the directed local Markov property is often simply referred to as the "Markov property for DAGs" in the literature.

The I-map, D-map and P-map definitions can also be extended to use the new $d$-separation criterion in DAGs.

**Definition 3.1.6 (Directed I-map)** *A DAG $\mathcal{G}$ is a* directed I-map *of a joint probability distribution $p(\cdot)$ if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \big( \mathrm{dSep}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \implies (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \big). \tag{3.10}$$

**Definition 3.1.7 (Directed D-map)** *A DAG $\mathcal{G}$ is a* directed D-map *of a joint probability distribution $p(\cdot)$ if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \big( \mathrm{dConn}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \implies (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \big). \tag{3.11}$$

**Definition 3.1.8 (Directed P-map)** *A DAG $\mathcal{G}$ is a* directed P-map *of a joint probability distribution $p(\cdot)$ when it is both a directed I-map and a directed D-map, i.e., if and only if:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} : \big( \mathrm{dSep}(X, Y \mid \mathbf{Z})_{\mathcal{G}} \iff (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \big). \tag{3.12}$$

Here again, not all distributions can be represented by directed P-maps. If a given distribution $p(\cdot)$ can be represented by a directed P-map, it is called *DAG-isomorphic*; in that case, at least one P-map exists—it is not unique in the general case.

Up to now, we have defined I-maps, D-maps and P-maps with respect to probability distributions. We can extent this to define them with respect to a dataset $D = \{\mathbf{v}^i \mid 1 \leq i \leq p\}$ with $\mathbf{v}^i$ assumed to be i.i.d. with respect to an unknown distribution $p(\mathbf{V})$. Conditional independencies are then found out with statistical tests. See Subsection 4.6.3 for a discussion of this issue.

### 3.1.3 More on Graph Isomorphisms

Some distributions are only DAG-isomorphic, some are only undirected graph-isomorphic, some are both and some are neither. A distribution that is neither directed graph-isomorphic nor undirected graph-isomorphic can be represented neither by directed nor undirected P-maps. It can, however, be represented by (un)directed I- or D-maps. If we had to choose, we would opt for the lesser of two evils: I-maps will fail to represent some independencies, D-maps will fail to represent some dependencies. Depending on the context, the one or the other can be preferred.[2]

[2]The concept of I- and D-maps comes from the Bayesian network terminology (see Subsection 3.2.1). In case of a Bayesian network, the more useful representation of a non-graph-isomorphic distribution is an I-map, because what we

Figure 3.2 shows a situation where we have $(X \perp\!\!\!\perp Y \mid \{Z, W\})$ and $(Z \perp\!\!\!\perp W \mid \{X, Y\})$, which cannot be represented by a DAG but can be represented by an undirected graph. Example data actually generating these independencies can be found in Moussouris (1974).

Figure 3.3 shows the classical V-structure, which is a DAG-isomorphic scenario but not undirected graph-isomorphic, because it violates the strong union property with $(X \perp\!\!\!\perp Y)$ and $(X \not\perp\!\!\!\perp Y \mid Z)$.

Finally, Figure 3.4 represents the only independency $(X \perp\!\!\!\perp Y \mid Z)$, which is both DAG- and undirected graph-isomorphic. For an example of a distribution which is neither DAG- nor undirected graph-isomorphic, see Subsection 4.1.2.

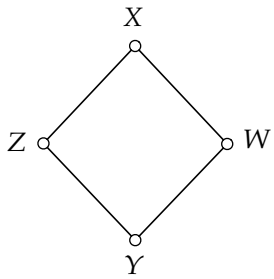



FIG. 3.2: Cyclic dependencies

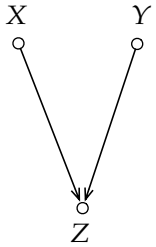



FIG. 3.3: V-structure

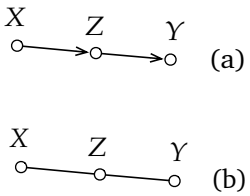



FIG. 3.4: Chained causation

If a distribution is DAG-isomorphic and not undirected graph-isomorphic, there is a simple way to get an undirected I-map out of the directed P-map $\mathcal{G}$ by taking the *moral graph* $\mathcal{G}^m$ of $\mathcal{G}$. The moral graph is constructed by creating new edges between all parents of the nodes with more than one parent and undirecting the remaining arcs. Note that certain independencies are lost when "marrying" a variable's parents; in that case, any directed P-map is transformed into an undirected I-map, losing the D-map property.

### 3.1.4 Independence Equivalence

In undirected graphs, the separation criterion makes use of the presence or absence of edges, which is what we expect. In DAGs, the $d$-separation criterion makes use of the presence of absence of arcs, but only looks at the direction of the arcs when it comes to converging nodes—the non-converging nodes can be either nodes in a directed chain or diverging nodes (i.e., a common causes for several effects).

Therefore, a set of conditional independencies will not fully determine a DAG, because the direction of some arrows will be left free. This particularity of the $d$-separation criterion is to be related to the causal underidentification problem described in Subsection 2.3.3. Two different DAGs—or, more generally, two different graphs—can thus represent the same set of conditional independencies. The concept of independence equivalence captures this idea.

**Definition 3.1.9 (Independence Equivalence)** *Two graphs $\mathcal{G}_1$, $\mathcal{G}_2$ are* independence equivalent

ultimately want to do is reduce the number of parameters of the distribution by showing the independence relations. A minimal I-map will thus be the most suited representation, allowing each independency represented in the graph to be found out in the distribution, without guaranteeing that *all* independencies are shown, whereas a D-map would fail to allow reconstruction of the full joint distribution.

*or* Markov equivalent *if and only if:*

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{X} : \big( (\mathrm{d})\mathrm{Sep}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_{\mathcal{G}_1} \iff (\mathrm{d})\mathrm{Sep}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_{\mathcal{G}_2} \big), \tag{3.13}$$

*i.e., if they represent the same set of conditional independence statements.*

With the directed and undirected possibilities for the two graphs, we have three cases:

1. $\mathcal{G}_1$ and $\mathcal{G}_2$ are undirected. They are independence equivalent if and only if they are identical.
2. $\mathcal{G}_1$ and $\mathcal{G}_2$ are DAGs. They are independence equivalent if and only if the following conditions are satisfied:
   - Their undirected version is identical;
   - The V-structures of both graphs are identical.
3. $\mathcal{G}_1$ is undirected and $\mathcal{G}_2$ is a DAG. They are independence equivalent if and only if the following conditions are satisfied:
   - The undirected version of the DAG $\mathcal{G}_2$ is identical to its moral graph $\mathcal{G}_2^m$;
   - The undirected version of the DAG $\mathcal{G}_2$ is identical to the undirected graph $\mathcal{G}_1$;
   - The undirected graph $\mathcal{G}_1$ is decomposable (see definition below).

   These conditions ensure that neither $\mathcal{G}_1$ nor $\mathcal{G}_2$ represents certain independence relations that cannot be represented by the other kind of graph.

**Definition 3.1.10 (Graph Decomposability)** *An undirected graph $\mathcal{G}$ is said to be* decomposable *if it is complete, or if there exists a decomposition $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ into decomposable subgraphs $\mathcal{G}_{\mathbf{X}\cup\mathbf{Z}}$ and $\mathcal{G}_{\mathbf{Y}\cup\mathbf{Z}}$. A triple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ of disjoint, nonempty subsets of nodes is said to form a* decomposition *of $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ if and only if the following conditions hold:*

- $\mathbf{V} = \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$*;*
- $\mathrm{Sep}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_{\mathcal{G}}$*, i.e., $\mathbf{X}$ and $\mathbf{Y}$ are separated by $\mathbf{Z}$;*
- *The induced subgraph $\mathcal{G}_{\mathbf{Z}}$ is complete.*

The definition of graph decomposability, a concept from graph theory, is only mentioned for the completeness of the definition of independence equivalence; it is not further discussed here.

To handle the particular problem of independence equivalence, other kinds of graphs can be used, like partially directed acyclic graphs (PDAGs), where some links can remain undirected. Maximally-oriented PDAGs, also called *completed* PDAGs or CPDAGs, are often used to represent an equivalence class found by some graph construction algorithms.

Consider the graphical example in Figure 3.5, and suppose the true causal structure of some process is (a). The set of conditional independencies drawn from it does not allow us to distinguish between (a), (b) and (c); they actually all belong to the equivalence class shown in the CPDAG (d).

One could wonder how the arc $Y \to U$ was directed in the CPDAG (d): there is no justifying V-structure for it. Actually, it is the *absence* of a V-structure involving it that unambiguously dictates the orientation in this case.

Figure 3.6 shows graphs which are not in this equivalence class. (a) has a reversed arc $U \to Y$ and thus two extra V-structures $Z \to Y \leftarrow U$ and $W \to Y \leftarrow U$. (b) has no reversed arc, but has

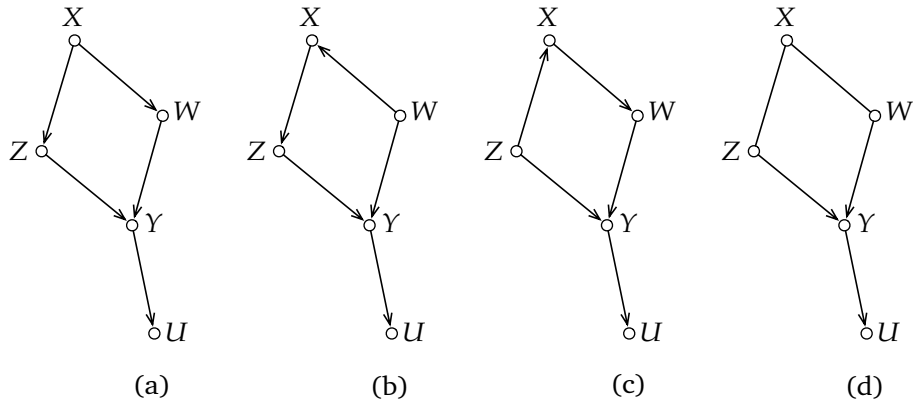



FIG. 3.5: DAGs (a), (b), (c), belonging to the equivalence class of the CPDAG (d)

an extra V-structure $Z \rightarrow X \leftarrow W$. (c) has the missing V-structure $Z \rightarrow Y \leftarrow W$ and is not even a DAG, as it contains a cycle.

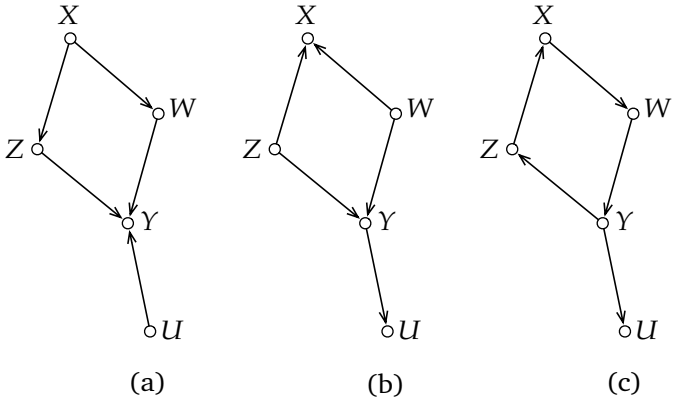



FIG. 3.6: Graphs not in the equivalence class shown in Figure 3.5 (d)

Yet another kind of graph, called *ancestor graph*, has been designed to allow for the representation of possible latent variables, i.e., hidden common causes in our model (Richardson & Spirtes, 2002). Ancestor graphs can contain undirected edges, directed arcs and bidirected links between variables. A generalization of the *d*-separation criterion called *m-separation* is then used.

Our discussion has now led us to consider that although DAGs and their extensions cannot represent any kind of possible set of conditional independencies, it is a much better way to model our causal graph than undirected graphs, which do not allow to adequately represent two causes for a single effect. We thus assume that the causal problems we will have to solve have a DAG-isomorphic structure. The issue of equivalence classes and of possible latent variables has led to the consideration of PDAGs and ancestral graphs, respectively, which both hint at which parts of the graph can or cannot be directed with the available information. While we will still represent ideal causal graphs with DAGs, we cannot expect the output of the graph construction algorithms to be more detailed than the CPDAG representing the equivalence class of the original DAG.

## 3.2 COMPARISONS & PARALLELS

So far, we have tried to define causal graphs intuitively, motivating our approach as further concepts were introduced. This introduction would be incomplete if it did not mention the parallel to Bayesian networks—actually, many of the reviewed concepts come from Bayesian network theory. We then briefly discuss Structural Equation Modeling, which is a well-known tool in social and economical sciences, and which has been used to model causation.

### 3.2.1 Bayesian Networks

A Bayesian network (Pearl, 1988) is a mathematical tool to effectively represent a joint probability distribution graphically, and to allow inference to be computed using its structure in a less computationally expensive way than directly reasoning with the joint distribution. When conditional independencies are found between some variables, the model can be simplified and the corresponding arcs removed.

This section is in no way intended to be a comprehensive introduction to Bayesian networks. The interested reader is invited to look at Charniak (1991) or Neapolitan (2003). We simply aim at underlining the similarities and differences between a causal network and a Bayesian network.

**Definition 3.2.1 (Bayesian Network)** *A* Bayesian network *(or* belief network*)* $\mathcal{B} = (\mathcal{G}, \mathbf{P})$ *consists of a directed acyclic graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *and of a set of conditional probability distributions* $\mathbf{P}$*, one for each node* $X \in \mathbf{V}$*, of the form* $p(X \mid \mathbf{Pa}(X))$.

The full joint distribution can then be expressed as the product of the conditional distributions at each node:

$$p(\mathbf{V}) = \prod_{X \in \mathbf{V}} p(X \mid \mathbf{Pa}(X)). \tag{3.14}$$

Note that any probability distribution can be represented by a Bayesian network. Using the chain rule, we know that in general:

$$p(X_1, X_2, \cdots, X_n) = p(X_1) \cdot p(X_2 \mid X_1) \cdots p(X_n \mid X_1, X_2, \cdots, X_{n-1}), \tag{3.15}$$

which is an equation of the form of Equation 3.14 with $\mathbf{Pa}(X_i) = \{X_k \mid 1 \leq k \leq i-1\}$. Graphically, it corresponds to a full graph. This is shown for $n = 4$ in Figure 3.7.

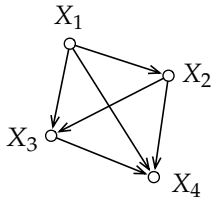
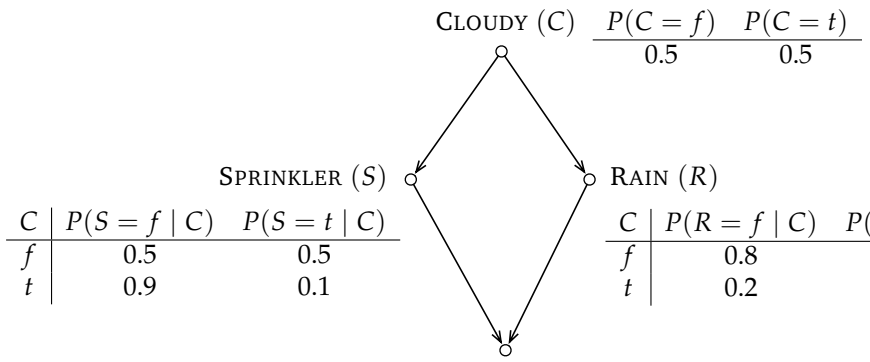



FIG. 3.7: A Bayesian network DAG able to represent any 4-variable distribution

But fully-connected graphs are not very informative nor useful. They assume that there is no conditional independency between the variables, and they actually do not simplify algebraical

reasoning over the joint distribution. The real benefits of Bayesian networks can be seen when their graphs are sparser. In Figure 3.8, we present a classical example, including the conditional probability tables. It describes whether or not my grass is wet, depending on the state of my sprinkler and on whether it is raining—two variables which in turn we assume depend on whether the sky is cloudy.




| $C$ | $P(C=f)$ | $P(C=t)$ |
|---|---|---|
| | 0.5 | 0.5 |

| $C$ | $P(S=f \mid C)$ | $P(S=t \mid C)$ |
|---|---|---|
| $f$ | 0.5 | 0.5 |
| $t$ | 0.9 | 0.1 |

| $C$ | $P(R=f \mid C)$ | $P(R=t \mid C)$ |
|---|---|---|
| $f$ | 0.8 | 0.2 |
| $t$ | 0.2 | 0.8 |

| $S$ | $R$ | $P(W=f \mid S,R)$ | $P(W=t \mid S,R)$ |
|---|---|---|---|
| $f$ | $f$ | 1.0 | 0.0 |
| $t$ | $f$ | 0.1 | 0.9 |
| $f$ | $t$ | 0.1 | 0.9 |
| $t$ | $t$ | 0.01 | 0.99 |

FIG. 3.8: A fully-specified Bayesian network

The $d$-separation criterion allows us to read conditional independencies off the graph, like $(S \perp\!\!\!\perp R \mid C)$, $(C \perp\!\!\!\perp W \mid \{S,R\})$ or $(S \not\perp\!\!\!\perp R \mid \{C,W\})$. Here, the joint distribution is

$$p(C,S,R,W) = p(C) \cdot p(S \mid C) \cdot p(R \mid C) \cdot p(W \mid S,R), \tag{3.16}$$

and we can clearly see the identified independencies which do not appear in the respective conditional distributions with respect to Equation 3.15: $S$ is absent from the conditional distribution of $R$, and $C$ is absent from that of $W$.

Specifying the full joint distribution for $n$ binary variables requires $2^n - 1$ parameters (the last one being determined by the constraint $\sum_{\mathbf{v}\in\mathcal{V}} p(\mathbf{v}) = 1$). In this case, 15 parameters. Identifying conditional independencies can reduce it to 9. This might not seem drastic here, but in general, the number of parameters grows exponentially for the joint distribution, whereas it is $\mathcal{O}(2^k \cdot n)$ for a Bayesian network with an average number of parents $k$.

Inference mechanisms using this structure would for instance answer queries such as "what is the probability that is is raining if I observe that the grass is wet" or "what is the probability that the sprinkler is off if it is not raining and no clouds can be seen." More generally, we want to know the specific distribution of a variable $X$ given the observational state of other variables, which are called *evidence* $\mathbf{E}$:

$$p(X \mid \mathbf{E} = \mathbf{e}). \tag{3.17}$$

Other possible queries include problems where we want to know the most likely state of the system

$\mathbf{x}^*$ given some evidence:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{X} = \mathbf{x} \mid \mathbf{E} = \mathbf{e}), \tag{3.18}$$

where $\mathbf{X}$ is a subset of the non-evidence variables $\mathbf{V} \setminus \mathbf{E}$.

Looking again at Figure 3.8, we could wonder what makes this example a Bayesian network rather than a causal network. Actually, in this configuration, it can be interpreted as a causal network, but in general, Bayesian network do not carry a causal semantics. If the topological order of variables in the graph is poorly chosen, we still obtain valid Bayesian networks, as shown in Figure 3.9 (albeit with more parameters and modified conditional probability tables), while the causal interpretation is obviously not applicable.

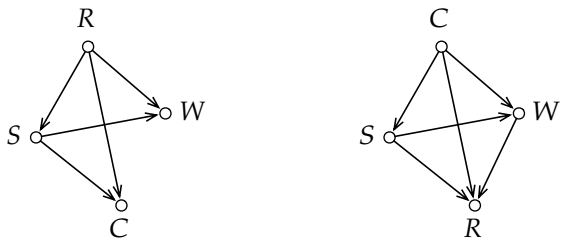



FIG. 3.9: Badly-chosen DAGs to represent the data from Figure 3.8

Thus, a causal network, provided it satisfies the I-map property and augmented by conditional probability tables, is always a valid Bayesian network, but the converse is not true. Several assumptions or properties related to causation are needed for constructing a causal network; see Section 4.3.

Bayesian network together with inference methods are good tools for computing marginal distributions given some evidence (although this is, in general, a NP-hard problem, several approximations exists: Gibb's sampling, likelihood weighting, variational methods; and exact polynomial algorithms have been developed for the case of singly connected graphs, also known as *polytrees*). Causal networks, together with the *do* calculus, on the other hand, aim at predicting the effects of interventions on the modeled system. The approach is completely different, although the resulting models have many similarities.

### 3.2.2 Structural Equation Models

In these models, causation between variables is expressed by a set of functions which deterministically determine the value of variables up to some disturbance term. This technique is called *functional causal models* in Pearl (2000), which is said to be more general than a Bayesian network with causal interpretation of the arcs.

For each variable, we have:

$$x_i = f_i(\mathbf{pa}_i, \varepsilon_i) \tag{3.19}$$

where $\mathbf{pa}_i$ is the set of variables used to determine $x_i$, and $\varepsilon_i$ is a stochastic zero-mean disturbance

(or error) term. A special case is the linear function

$$x_i = \langle \boldsymbol{\beta}_i, \mathbf{pa}_i \rangle + \varepsilon_i \tag{3.20}$$

where $\boldsymbol{\beta}_i$ is the coefficient vector. In matrix notation, we have, for the linear case:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \boldsymbol{\varepsilon} \tag{3.21}$$

where $\mathbf{B} = \begin{pmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \cdots & \boldsymbol{\beta}_n \end{pmatrix}^T$ is the coefficient matrix. If we represent it graphically, drawing an arc from parents to children, we obtain, once again, a causal graph, if we have no causal loop. This leads us to the condition that, for each acyclic model, there exists an ordering over the variables such that the coefficient matrix $\mathbf{B}$ is lower-triangular with a zero diagonal.

Structural equations carry causality in their definition (contrary to the tendency among economists and social scientists), namely, the only causes for $X_i$ are $\mathbf{Pa}(X_i)$ and no other variable modeled in the set of equations. The possible effect of variables that we do not want to or cannot take into account and that are thus not being modeled can be included in the disturbance term $\varepsilon_i$.

Whenever two disturbance terms $\varepsilon_i, \varepsilon_j$ have a nonzero covariance, this is a hint at a latent common cause. The intuition of trying to look for correlation in the error terms is discussed at length in Section 5.1. Models are called *Markovian* if they are acyclic and have independent error terms, or *semi-Markovian* if the latter assumption does not hold. Semi-Markovian models can be made Markovian by introducing latent variables to explain the correlation between the error terms.

Using the *do* notation, $y = f(x, \varepsilon)$ says that $f(x, \varepsilon)$ is the expected value of $Y$ when we set $X = x$, namely $\mathbb{E}(Y \mid do(X = x))$ and *not* $\mathbb{E}(Y \mid X = x)$. The equality sign $=$ in the functional equations should be interpreted as *assignment* $:=$ rather than symmetrical mathematical equality, reflecting the asymmetry of causation. As these equations represent the physical/natural process of causation, they cannot be reversed or used as a traditional equation set in a non-paradoxical way. If we define

$$y = \beta x + \varepsilon, \tag{3.22}$$

then $\beta$ is interpreted as the change in $\mathbb{E}(Y)$ per unit change of $X$. We cannot write, however,

$$x = \frac{y - \varepsilon}{\beta}, \tag{3.23}$$

since then $1/\beta$ would be interpreted as the change in $\mathbb{E}(X)$ per unit change of $Y$, which does not hold since $X$ was defined as a cause for $Y$. $Y$ could very well be artificially set to $y$ with no influence on $X$. This imposes a directed way to read the structural equations.

Structural equations also describe a joint probability distribution. Although the equations are deterministic, the disturbance terms can create the randomness needed to simulate the stochastic part.[3] If the model is linear and given a prior ordering on the variables, the $\beta$ coefficients can be estimated with the least-squares technique.

Like in Bayesian networks, DAGs are used to represent the dependencies between the variables. The bonus of replacing the conditional probability distributions by the $f_i$ functions is that we can now answer additional types of queries known as *counterfactuals*. For instance, referring to the

[3] Pearl (2000) also proves that SEM is equivalent to the Neyman–Rubin potential outcome framework, "understood by few and used by even fewer."

example pictured in Figure 3.8: "would the grass be wet if the sprinkler had been off, given that the grass is actually not wet and the sprinkler is on?" The interested reader is invited to read Chapters 7 and 8 of Pearl (2000) to read more about counterfactuals.

## 3.3 EFFECT OF INTERVENTIONS

Let us further explore the difference between observation and intervention by building on the proposed solution to Simpson's paradox mentioned in Subsection 2.1.2. In particular, we will review the rules of the *do* calculus and show how they lead to the aforementioned result.

### 3.3.1 Manipulated Graphs

By intervening on a system, we want to achieve a certain effect on some variables (that possibly cannot be tuned directly) by setting other, more accessible variables. After an initial study conducting to a causal graph, intervening on a system amounts to changing the graph structure, changing the conditional probability tables (in terms of Bayesian networks) or changing the structural equations (in terms of SEM).

Let $X$ be a tunable variable and $Y$ a response variable somehow causally influenced by $X$. We can distinguish several types of interventions, in increasing levels of complexity, along with their notation:

1. Unconditionally setting $X$ to some value $x$. We then observe $Y$: $p(Y \mid do(X = x))$;
2. Setting $X$ to some value, depending on some control variable $Z$: $p(Y \mid do(X = x \mid Z = z))$, or more generally, $p\big(Y \mid do(X = g(z))\big)$;
3. Replacing the original conditional distribution of $X$ (respectively, its structural equation) by a new distribution or equation, potentially with a new set of parents: $p\big(Y \mid do(p'(X \mid \mathbf{Pa}(X)'))\big)$ or $p\big(Y \mid do(x = f'_X(\mathbf{pa}'_X, \varepsilon'_X))\big)$.

Intervention 1 is a special case of Intervention 2, which is itself a special case of Intervention 3.

Interventions 2 and 3 are subject to the acyclicity constraint: $Z \notin \mathbf{De}(X)$ and $\mathbf{Pa}(X)' \cap \mathbf{De}(X) = \emptyset$, respectively. Realizing that they create causal dependencies between $Z$ and $X$, and $\mathbf{Pa}(X)'$ and $X$, respectively, we must see to it that we do not create causal feedback loops, forbidden by our model based on DAGs.

Suppose, looking at Figure 3.8 again, that we perform the following interventions on the system, corresponding to the three types of interventions we identified:

(a) We decide to turn on the sprinkler;
(b) We turn on the sprinkler if it is not raining;
(c) Depending on RAIN and on a new variable TEMPERATURE ($T$)—which we suppose independent from CLOUD and RAIN—we turn it off or on.

Figure 3.10 shows what are called *manipulated graphs* for these three scenarios. The variable we intervene on is marked with a bullet (•).

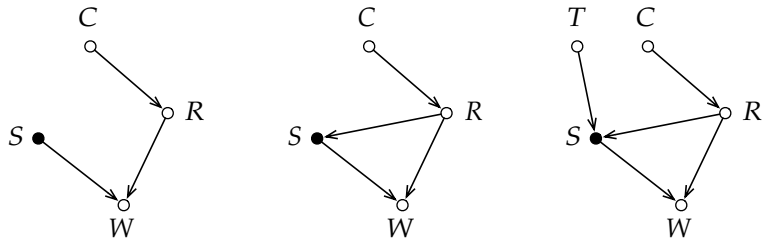



FIG. 3.10: The manipulated graphs

Now, we can examine the effect of an atomic intervention of the type of (a), $p(W \mid do(S = on))$: the incoming arcs of $S$ are suppressed, i.e., setting $S$ affects only its effect and not its cause:

$$p(W) \neq p(W \mid do(S = on)) \tag{3.24}$$
$$p(C) = p(C \mid do(S = on)) \tag{3.25}$$
$$p(R) = p(R \mid do(S = on)). \tag{3.26}$$

On the contrary, observing $S$, as in $p(W \mid S = on)$, also affects the probability distribution of its cause, and thus of the other effect of its cause:

$$p(W) \neq p(W \mid S = on) \tag{3.27}$$
$$p(C) \neq p(C \mid S = on) \tag{3.28}$$
$$p(R) \neq p(R \mid S = on). \tag{3.29}$$

The new joint probability distribution resulting from an intervention is called *postintervention* distribution, as opposed to the *preintervention* distribution. The following equation shows how to compute them in case (a), which will be the only one discussed here. For details on more complicated interventions, refer to Hagmayer *et al.* (2005), Woodward (2003) or to Chapter 4 of Pearl (2000).

$$p(\mathbf{V} \mid do(X = x)) = \begin{cases} \prod_{V \in \mathbf{V}, V \neq X} p(V \mid \mathbf{Pa}(V)) & \text{if } X = x, \\ 0 & \text{if } X \neq x. \end{cases} \tag{3.30}$$

Manipulated graphs are used in the rules of the *do* calculus. With $X, Y \in \mathbf{V}$, the following manipulated graphs are defined:

- $\mathcal{G}_{\overline{X}}$ is the graph where all arcs going into $X$ have been deleted (we suppress the influence of the causes for $X$);
- $\mathcal{G}_{\underline{X}}$ is the graph where all arcs going out of $X$ have been deleted (we suppress the influence of $X$ on its effects).

### 3.3.2 *do* Calculus

The *do* calculus is originally described in Pearl (1995), where proofs are given for the rules, using manipulated graphs. It describes how to turn a *do* expression into an expression involving only observable quantities.

**Definition 3.3.1 (Rules of *do* Calculus)** *For all (disjoint subsets of) variables $X, Y, Z, W \in \mathbf{V}$ (respectively, $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W} \subset \mathbf{V}$) in a causal graph $\mathcal{G}$, we have the following rules.*

- **Rule 1: Insertion and deletion of observations**

$$\mathrm{dSep}(Y, Z \mid X \cup W)_{\mathcal{G}_{\overline{X}}} \implies p(y \mid do(x),\, z,\, w) = p(y \mid do(x),\, w) \tag{3.31}$$

- **Rule 2: Action/observation exchange**

$$\mathrm{dSep}(Y, Z \mid X \cup W)_{\mathcal{G}_{\overline{X}\underline{Z}}} \implies p(y \mid do(x),\, do(z),\, w) = p(y \mid do(x),\, z,\, w) \tag{3.32}$$

- **Rule 3: Insertion and deletion of actions**

$$\mathrm{dSep}(Y, Z \mid X \cup W)_{\mathcal{G}_{\overline{X},\overline{Z \setminus \mathbf{An}(W)}}} \implies p(y \mid do(x),\, do(z),\, w) = p(y \mid do(x),\, w) \tag{3.33}$$

Rule 1 establishes an equality between two probabilities where the only difference lies in an additional variable $Z$ being conditioned on, and gives a familiar $d$-separation condition, but in the manipulated graph. It reaffirms the graph manipulation consisting of removing the incoming arcs for $X$ as the correct representation of a single intervention $do(X = x)$ and maintains $d$-separation as a valid criterion for reading conditional independencies off the graph.

Rule 2 gives a condition for the equivalence of doing and seeing $Z = z$ as far as the conditional distribution of $Y$ is concerned: $Z$ and $Y$ must be $d$-separated in the graph where all outgoing arcs from $Z$ have been deleted. This is related to what is known as the *back-door criterion* and allowed us to solve our problem resulting from Simpson's paradox. To exchange observation for intervention, we require all influences from $Z$ on $Y$ to go through the direct effects of $Z$ (and hence, $Z$ and $Y$ to be $d$-separated once we remove the effect of $Z$ on its children). Otherwise, we are in a situation similar to the one in Figure 2.1 and we must adjust for the confounding variables by conditioning on them (or by setting them to certain values). Thus they appear in the conditioning set of the $d$-separation condition of Rule 2, $d$-separate $Z$ from $Y$ and in this context allow the equivalence of observation and intervention.

Finally, Rule 3 tells when adding an intervention on $Z$ has no effect on a variable $Y$: namely, when $Z$ and $Y$ are $d$-separated in the manipulated graph, just like $S$ and $C$, respectively, in Figure 3.10 (a). If $X$ is additionally being set, then we must manipulate the graph accordingly. The reason for limiting the arc deletion to nonancestors of $W$ can be found in Pearl (1995).

These rules, although not straightforward, can be understood intuitively by visualizing the graphs and can be applied systematically to solve a *do* expression. We do not further detail their use here as we turn to the main topic of this study, namely causal graph structure learning, in the next chapter.

**Summary**

*In this chapter, we saw that directed acyclic graphs are the best way to graphically represent a causal graph, although we know that some particular cases cannot be faithfully represented by them. The Markov conditions and the d-separation criterion link graphs to probability distributions. Whether we interpret the causal graph to be a special case of a Bayesian network or a visual proxy for a set of structural equations, we can use it to compute the effects of interventions with the* do *calculus, manipulating or transforming the graph as needed according to a set of rules.*

CHAPTER

4

# Structure Learning Algorithms

*We now turn to structure learning algorithms. As our causal networks are close to Bayesian networks, we first look at various traditional Bayesian network construction algorithms to draw inspiration from them. We then review the main causal network construction algorithms and their assumptions. We distinguish for now two main kinds of algorithm, one based on a search-and-score approach and one constructing the structure under the conditional independence constraints. Each algorithm is tested with a toy example and three dataset sizes and a non-DAG-isomorphic dataset representing a* XOR *problem. Finally, the non-straightforward steps are detailed and explained.*

## 4.1 TEST DATASETS

First, we introduce the two toy datasets with which each algorithm was tested.

### 4.1.1 Toy Retail Data

Data points are sampled from a predefined Bayesian network with the structure represented in Figure 4.1. Using a well-defined Bayesian network and sampling from it ensures that the obtained dataset generates a DAG-isomorphic set of conditional independencies in the large-sample limit. This is a good example to check how well algorithms perform with respect to a known reference graph.

All six nodes represent discrete variables. The node AVAILABILITY has values *yes* or *no*. The node PROMOTION can be *on* or *off*. The SALES and the DEMAND can both be *low*, *medium* or *high*. The node SEASON is understandably one of the four seasons, and the WEATHER has been simplified to be either *sunny* or *raining*.

The sampling is independent and identically distributed according to the Bayesian network's conditional probability tables, which we do not detail here. We performed runs of the algorithms with sample sizes of 50, 200 and 1000, the results of which are above the letters (a), (b) and (c), respectively, for each of the following figures.

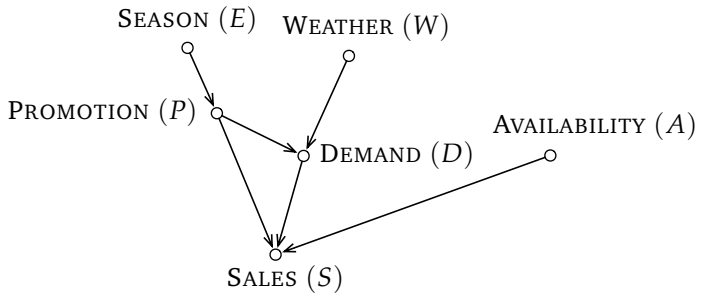



FIG. 4.1: The reference network structure of the Toy Retail dataset

This network is well-suited for algorithm comparison because:

- It is sparse enough to include interesting conditional independencies;
- It contains identifiable V-structures;
- Its corresponding equivalence class only contains two DAGs, such that the causal underidentification problem, although recognizable, will not be a major issue here.

The two members of the equivalence class are shown in Figure 4.2.

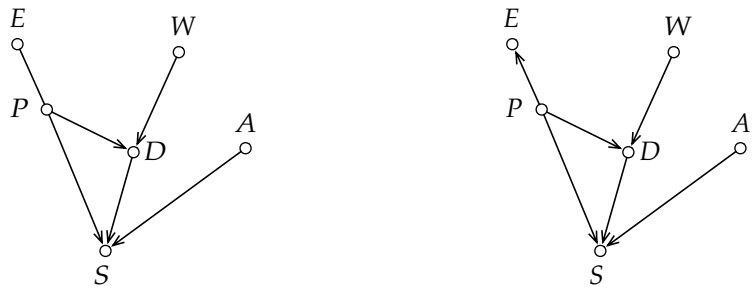



FIG. 4.2: (a) The equivalence class of the DAG in Figure 4.1 and (b) the other member of the class

### 4.1.2 XOR Problem

The previous example does not violate the assumptions of the tested algorithms, in particular the assumption that the problem has a DAG-isomorphic distribution. Let us look at an example involving data that does not generate a set of conditional independencies which can be faithfully represented by a DAG. For the sake of simplicity, we chose the XOR problem, also known as the 2-bit parity problem.

It is a problem with three binomial variables, two of which ($X$ and $Y$) are distributed uniformly and independently between 0 and 1, and the last of which ($Z$) is defined as $Z = X \oplus Y$, where $\oplus$ is the XOR operator. It has the interesting property that also $Y = X \oplus Z$ and $X = Y \oplus Z$. Note that in this example, when two variables have been set, the third one is determined deterministically, so that the only possible data points are $(X, Y, Z) = (0,0,0)$, $(0,1,1)$, $(1,0,1)$ and $(1,1,0)$.

Note that the distribution $p(X,Y,Z)$ is not strictly positive for all $(x,y,z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, such that the Intersection property of conditional independence does not hold any more. As far as the directed Markov properties are concerned, we lose the implication (DP) $\Rightarrow$ (DL).

In the large sample limit, the mutual information or correlation of any two of these variables will be zero. The following conditional independencies and dependencies hold:

$$\begin{array}{lll} (X \perp\!\!\!\perp Y) & (Y \perp\!\!\!\perp Z) & (X \perp\!\!\!\perp Z) \\ (X \not\perp\!\!\!\perp Y \mid Z) & (Y \not\perp\!\!\!\perp Z \mid X) & (X \not\perp\!\!\!\perp Z \mid Y) \end{array} \tag{4.1}$$

It is impossible to find a graph to depict all of these relations. Possible directed minimal I-maps are shown in Figure 4.3.

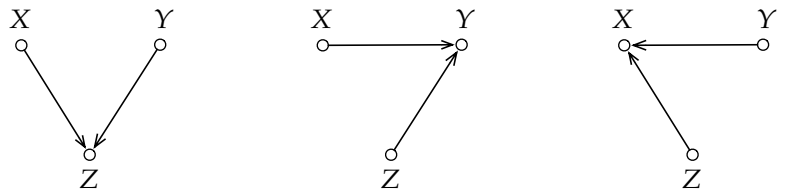



Fig. 4.3: Possible directed I-maps for the XOR problem

The XOR problem can be generalized to higher dimensions. Guyon & Elisseeff (2003) discuss it in the context of feature selection. Feature selection is a problem related to causal network construction, see Section 5.3 for a discussion. This dataset shows how much a violation of the DAG-isomorphicity assumption affects a given algorithm—whether it is still able to recover a graph close to an I-map or whether it fails completely for some reason. In the figures, the results of this test are shown above the letter (d).

The tests were run using specific Matlab toolboxes (Murphy, 2000; Leray & François, 2004a) and/or our own implementation on a 1 GHz machine.

## 4.2 Learning Bayesian Networks

Before looking at causal structure learning algorithms, we will review some common Bayesian network structure learning techniques.

### 4.2.1 Problem Statement

One is given a dataset $D = \{\mathbf{v}^i \mid 1 \leq i \leq p\}$ with $\mathbf{v}^i = (x_1^i, x_2^i, \cdots, x_n^i)^T$, assumed to be i.i.d. with respect to an unknown distribution $p(\mathbf{V})$. Data may be missing, in which case we have for some $i$ and $j$: $x_j^i = \text{n/a}$. The goal is to extract a Bayesian network $\mathcal{B} = (\mathcal{G}, \mathbf{P}_{\boldsymbol{\theta}})$ from $D$ such that the DAG $\mathcal{G}$ is as close as possible to a P-map for the distribution $p(\mathbf{V})$. $\boldsymbol{\theta}$ is a array of parameters describing the conditional probabilities for each node given its parents.

*Structure learning* is mainly about determining $\mathcal{G}$—determining $\boldsymbol{\theta}$ is called *parameter learning*. It turns out that several structure learning algorithms also estimate $\boldsymbol{\theta}$ to perform various goodness-of-fit tests on a given structure.

Structure learning is a difficult problem: for $n$ variables, the number of possible DAGs $\nu(n)$ grows superexponentially with $n$ according to the recursive formula (Robinson, 1977):

$$\nu(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} \cdot \nu(n-i). \tag{4.2}$$

We have: $\nu(2) = 3$, $\nu(3) = 25$, $\nu(5) = 29281$, $\nu(10) \approx 4.2 \times 10^{18}$.

The space of Markov equivalent DAGs, although smaller, also grows superexponentially with the number of variables. Needless to say, an exhaustive search is not a realistic approach with more than 5 or 6 nodes; we need more elaborate algorithms.

For Bayesian networks, we differentiate between algorithms able to work with a dataset containing missing values and those which require complete datasets. We first introduce *scoring functions* as a way to determine how good a given Bayesian network is for a certain dataset.

### 4.2.2 Scoring Functions

For a certain class of algorithms, we need a way to assess whether or not a given Bayesian network is good for our dataset. A possibility is to maximize the probability that the network has generated the given dataset by using this probability as score:

$$\text{score}(\mathcal{B}, D) = p(D \mid \mathcal{G}, \hat{\boldsymbol{\theta}}) \tag{4.3}$$

with $\hat{\boldsymbol{\theta}}$ being e.g. the maximum likelihood (ML) or maximum a posteriori (MAP) estimator of $\boldsymbol{\theta}$ (or a vector of estimators $\hat{\theta}_i$). The ML estimator is defined by:

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} p(D \mid \boldsymbol{\theta}). \tag{4.4}$$

The MAP estimator is not so dissimilar: it mainly includes the assumption that a prior distribution $p(\boldsymbol{\theta})$ of the parameters $\boldsymbol{\theta}$ exists.[1] The MAP estimator is:

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(D \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{4.5}$$

Another possible scoring function, known as Bayesian scoring, integrates over the parameters:

$$\text{score}_B(\mathcal{B}, D) = p(D \mid \mathcal{G}) = \int_{\boldsymbol{\theta}} p(D \mid \mathcal{G}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{G}) \, d\boldsymbol{\theta}. \tag{4.6}$$

The distribution $p(\boldsymbol{\theta} \mid \mathcal{G})$ represents the prior on $\boldsymbol{\theta}$. In the discrete case, each parameter $\theta_i$ from $\boldsymbol{\theta}$ is Dirichlet-distributed (or beta-distributed in the binomial case) and the integral can be written in closed form and efficiently computed without needing to enumerate all possible parameters.

It turns out that this simple score will encourage more complex networks with lots of dependencies in order to best fit the data. Although we do want to best fit the data, we want models whose com-

[1] When doing parameter learning, this can be useful if we want to inject prior knowledge into the network to be built. In structure learning, since we have to learn the structure and parameters exhaustively, we will most probably not have any prior distribution on potential parameters. As the number of samples $p$ in our dataset approaches infinity, $\hat{\boldsymbol{\theta}}_{MAP}$ tends to $\hat{\boldsymbol{\theta}}_{ML}$.

plexity is small enough to be interpreted easily and prevent overfitting. Therefore, the *Bayesian information criterion* (BIC) was defined:

$$\text{score}_{BIC}(\mathcal{B}, D) = \log p(D \mid \mathcal{G}, \hat{\boldsymbol{\theta}}) - \frac{1}{2} \dim(\mathcal{B}) \log p, \tag{4.7}$$

with $\dim(\mathcal{B})$ being the *dimension* of the network $\mathcal{B}$, i.e., generally the number of independent parameters describing it. This scoring function is intuitively understandable, the first term being the same as in the previous score, and the second term punishing the model for being too complex.

Whether a given scoring function is more appropriate than the others depends on the dataset.

A property of scoring functions called *decomposability* is useful for implementing efficient subsequent computations of scores of similar networks.

**Definition 4.2.1 (Decomposability)** *A scoring function is said to be* decomposable *if it can be written as the sum or product of local scoring functions $s(X_i, \mathbf{Pa}(X_i))$ depending only on a node $X_i$ and its parents $\mathbf{Pa}(X_i)$, e.g.:*

$$\text{score}(\mathcal{B}, D) = \sum_{i=1}^{n} s(X_i, \mathbf{Pa}(X_i)). \tag{4.8}$$

This property is useful when scoring networks which are similar to one another, because it gives a hint at what kind of computation can be reused in subsequent scoring and saves substantial processing time.

The Bayesian score is decomposable. Integrating over the Dirichlet-distributed MAP parameters conveniently yields the following (the Gamma function $\Gamma(\cdot)$ was replaced by factorials; $p_{ijk}$ is the number of occurrences of variable $X_i$ taking its $k$th value, $1 \leq k \leq r_i$ when its parents are in their $j$th configuration, $1 \leq j \leq q_i$):

$$\text{score}_B(\mathcal{B}, D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{\left(\sum_{k=1}^{r_i} p_{ijk} + r_i - 1\right)!} \prod_{k=1}^{r_i} p_{ijk}! \tag{4.9}$$

$$= \prod_{i=1}^{n} s_B(X_i, \mathbf{Pa}(X_i)). \tag{4.10}$$

The BIC score is also decomposable ($\boldsymbol{\theta}_{X_i|\mathbf{Pa}(X_i)}$ is the set of parameters required to describe the conditional distribution of $X_i$ given its parents):

$$\text{score}_{BIC}(\mathcal{B}, D) = \sum_{i=1}^{n} s_{BIC}(X_i, \mathbf{Pa}(X_i), \boldsymbol{\theta}_{X_i|\mathbf{Pa}(X_i)}) \tag{4.11}$$

$$s_{BIC}(X_i, \mathbf{Pa}(X_i), \boldsymbol{\theta}_{X_i|\mathbf{Pa}(X_i)}) = \sum_{x_k} \sum_{\mathbf{pa}_j} p_{ijk} \log p(\theta_{ijk}) - \frac{1}{2} \dim(\boldsymbol{\theta}_{X_i|\mathbf{Pa}(X_i)}) \log p. \tag{4.12}$$

### 4.2.3 Search-and-Score Algorithms with Complete Datasets

Let us now show some standard construction algorithms which use scoring functions. More details can be found in Leray & François (2004b).

### K2

The K2 algorithm, introduced by Herskovits (1991) and Cooper & Herskovits (1992), maximizes the structure probability given the data. It requires a prior ordering **o** on the nodes and will only look for parents for a specified node in its predecessors according to **o**, which is therefore a topological order of the nodes of the final graph. K2 thus requires expert knowledge or hints which can help define this order.

Starting with an empty graph, K2 looks at all potential parents of each node and adds an arc between them in the direction specified by **o** if the score increases. The algorithm is fast but neglects a big part of the total search space because of the constraints imposed by **o**. Obviously we cannot make it run with every possible **o**, whose number grows exponentially with $n$.

The constructed graphs are shown in Figure 4.4. Run times for (a), (b), (c) and (d) were 0.99 s, 1.00 s, 1.04 s and 0.19 s, respectively.

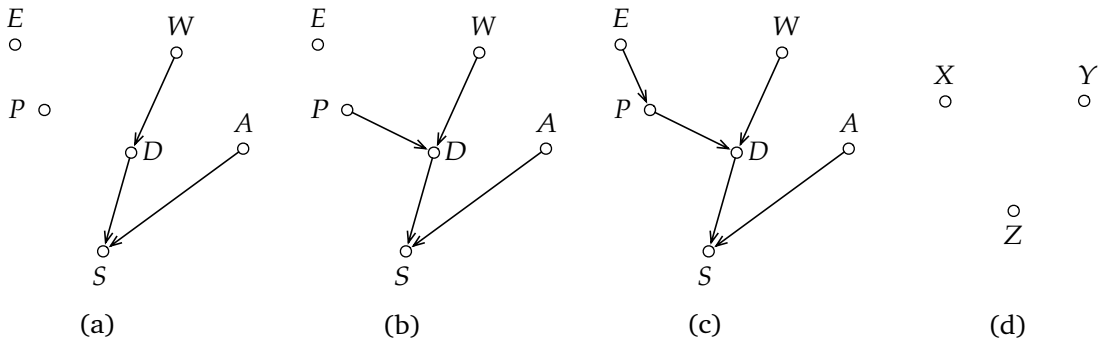



FIG. 4.4: Graphs for K2: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

K2 will not make arc orientation errors, as it orients them according to the prior ordering. For the Toy Retail problem, the results are pretty consistent—an arc added for a sample size of $p = p_1$ is kept in experiments where $p > p_1$. With $p = 1000$, only one arc is missing. This arc is actually missing in a lot of results and it actually corresponds to the weakest conditional association in the model.

K2 fails at the XOR test completely, however. In general, a pairwise examination of variables will be fatal in this problem, as no variable alone carries information about any other.

### Maximum Weight Spanning Tree (MWST)

The idea behind the MWST technique (Chow & Liu, 1968) is to assign a weight to each possible edge according to some similarity criterion and then build the maximum spanning tree of the obtained graph (using e.g. Prim's (Prim, 1957) or Kruskal's (Kruskal, 1956) modified version of the minimal spanning tree algorithm). The resulting tree is undirected and the root can then be chosen to obtain a directed graph.

The criterion used to determine the edge weights can for instance be the mutual information between the two variables or the score variation when this node is added. This is computed, however, when the graph is empty and it is not updated dynamically as the DAG is built: this is how it differs from a greedy search, for instance.

Note that by definition a node in a tree has one parent only. Therefore, a node cannot have more than one incoming link, and thus this algorithm has a limited practical use. (Note that this algorithm was introduced before Bayesian networks became known in the eighties.) It can be used, however, to find a prior ordering of the variables or an initial structure to initialize other algorithms. We still need to designate one variable as the root, i.e., to find one variable which we know should have no parents (or no observed cause).

Although the variable order returned by MWST can be used to initialize other algorithms, we should take care that this second algorithm does not hold MWST results for verified expert data, not allowing for modifications. While initializing a greedy search with the structure found by MWST seems to be a good idea, using a topological sort of the variables in MWST's graph as prior ordering for K2 seems somewhat arbitrary: the topological sort is unique only in the case of a chain of nodes and loses uniqueness as soon as some node has more than one child.

The constructed graphs are shown in Figure 4.5. Run times were 0.09 s, 0.13 s, 0.28 s and 0.20 s, respectively.

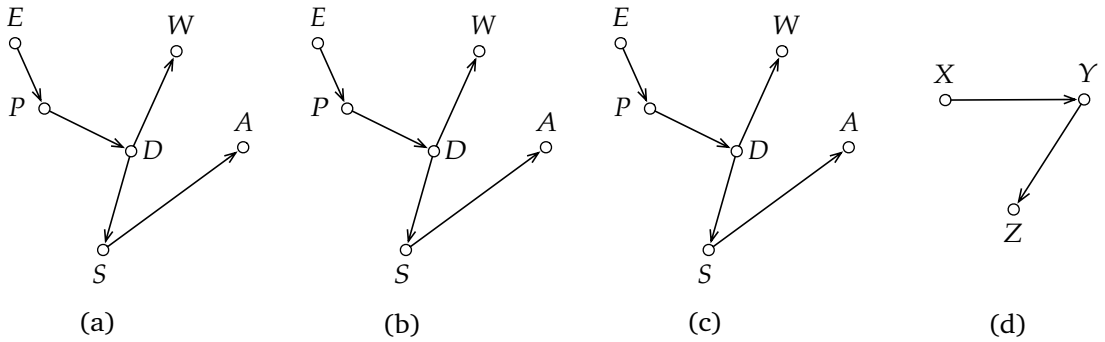



FIG. 4.5: Graphs for MWST: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

MWST yields robust results and consistently outputs the same graph for the Toy Retail test. It misses one arc (which is a constraint imposed by the definition of a tree) and misdirects two others (also because of the tree constraint). The undirected structure, however, is a tree-like skeleton of the initial structure.

A connected structure is found for the XOR problem, which is unfortunately not an I-map: we have $\mathrm{dSep}(X, Z \mid Y)$, but $(X \perp\!\!\!\perp Z \mid Y)$ does not hold. This has again to do with the tree constraint: any I-map for this problem is a V-structure and therefore not a tree. We see, however, that whereas other algorithms (typically, of the type Search-and-Score) miss links with the small dataset, or others (typically, of the type Construction under Constraints) include too many, MWST will always construct a tree singly connecting all nodes.

#### Greedy Search (GS)

The GS algorithm starts with an empty graph (or some other initial graph) and chooses a graph in its neighborhood which locally optimizes the score until no further improvement is possible. It is a hill-climbing approach.

The neighborhood of the graph is defined as all DAGs that can be obtained through arc addition, arc deletion or arc reversal with respect to the current graph. This might look like a lot to test and

the neighborhood indeed has a size of $\mathcal{O}(n^2)$, but the decomposability of the scoring functions as well as the caching of already-scored structures help make it faster.

Like all pure hill-climbing approaches, this greedy search suffers from the problem of getting stuck in a local optimum. Traditional techniques could be added to the basic algorithm, like multiple restarts with random initializations or some Simulated Annealing search behavior.

The constructed graphs are shown in Figure 4.6. Run times were 6.87 s, 9.59 s, 14.40 s and 1.17 s, respectively.

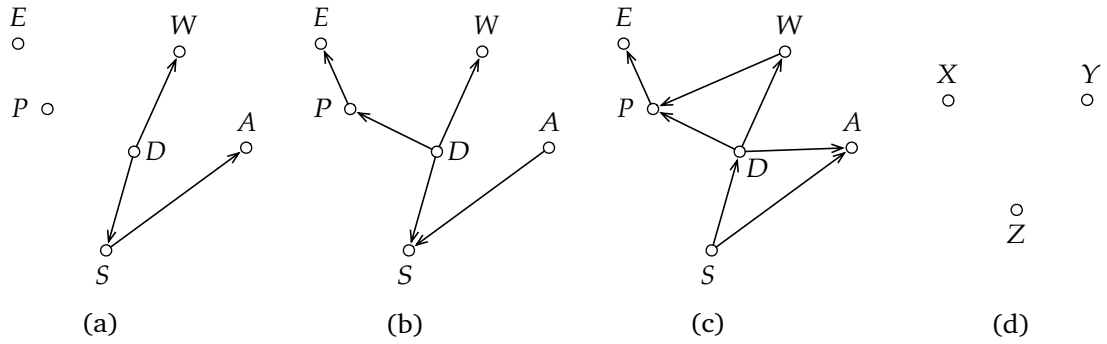



FIG. 4.6: Graphs for GS: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

The results of GS seem to be rather unpredictable. Across the three experiments with the Toy Retail dataset, only one arc can be systematically found—and it is misdirected with respect to the true structure. No luck with the XOR test either, which produces an empty graph.

#### Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain (Metropolis *et al.*, 1953). This chain is constructed such that its stationary distribution is the distribution we want to sample from. The state of the chain after a large number of steps is then used as a sample from the desired distribution. The quality of the sample improves as the actual distribution approaches the stationary distribution. Practically, we let the chain perform a few hundred transitions, known as *burn-in time* or *mixing time*, before considering drawing actual samples from it.

MCMC methods were originally designed to calculate the ergodic limit for models of physical systems. The trivial solution is to run the model for a sufficiently long time and then average over time. The idea behind MCMC is that any other Markov process which has the same ergodic limit will also do, and Metropolis *et al.* (1953) provided an algorithm for constructing such a Markov chain.

We recall that the stationary distribution $\boldsymbol{\pi}$ associated with a Markov chain is the eigenvector corresponding to eigenvalue 1 of the conditional probability transition matrix $\mathbf{P}$:

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi} \tag{4.13}$$

$$\operatorname{diag}\left(\lim_{k\to\infty} \mathbf{P}^k\right) = \boldsymbol{\pi} \tag{4.14}$$

There is a unique solution for $\boldsymbol{\pi}$ only if the Markov chain is ergodic, i.e., if its transition matrix is irreducible (any state can be reached from any other state) and aperiodic (there is no state to which the process will continually return with a fixed time period).

The implemented algorithm is a special case of the *Metropolis-Hastings algorithm* (MH) as generalized by Hastings (1970). It works on the space of DAGs; i.e., the states of the Markov chain each represent a valid DAG. It assesses a transition probability from a state to the other using the respective Bayes scores of the corresponding DAGs. This method is nondeterministic. Its output can either be the highest-scored DAG or a set of good DAGs suitable to perform model averaging.

Of course, we cannot keep in memory a whole Markov chain representing all possible DAGs. Therefore, we only store the states which are actually visited. Transitions from one state $\mathcal{G}$ to another state $\mathcal{G}'$ can happen if the two corresponding DAGs are in the neighborhood of one another, and happens, according to MH, with the following probability:

$$P(\mathcal{G} \to \mathcal{G}') = \min\left\{1, \frac{P(\mathcal{G}' \mid D)Q(\mathcal{G}' \to \mathcal{G})}{P(\mathcal{G} \mid D)Q(\mathcal{G} \to \mathcal{G}')}\right\}. \tag{4.15}$$

$Q(\mathcal{G} \to \mathcal{G}')$ can be seen as our "transition policy" when going from state $\mathcal{G}$ to state $\mathcal{G}'$. It could for example be evenly distributed for each of the basic operations: arc deletion, insertion and reversal (as long as it still produces a valid DAG). The implementation always has $Q(\mathcal{G} \to \mathcal{G}') = Q(\mathcal{G}' \to \mathcal{G})$ such that they cancel out in the transition probability. Using Bayes' rule, we have $P(\mathcal{G} \mid D) = \alpha P(D \mid \mathcal{G})P(\mathcal{G})$, where $\alpha$ is a normalizing constant independent of $\mathcal{G}$. If we have uniform priors on the DAGs, then:

$$\begin{aligned} P(\mathcal{G} \to \mathcal{G}') &= \min\left\{1, \frac{P(\mathcal{G}' \mid D)}{P(\mathcal{G} \mid D)}\right\} \\ &= \min\left\{1, \frac{\alpha P(D \mid \mathcal{G}')P(\mathcal{G}')}{\alpha P(D \mid \mathcal{G})P(\mathcal{G})}\right\} \\ &= \min\left\{1, \frac{P(D \mid \mathcal{G}')}{P(D \mid \mathcal{G})}\right\}. \end{aligned} \tag{4.16}$$

$P(D \mid \mathcal{G}')/P(D \mid \mathcal{G})$ is called *Bayes factor* (or *weight of evidence*) and provides a measure of "how better" a model is with respect to another. It can be computed efficiently thanks to its decomposability.

Despite these efficiency improvements, the MH algorithm is rather slow compared to others. The constructed graphs are shown in Figure 4.7. Run times were 10.64 s, 11.52 s, 14.36 s and 12.09 s, respectively.

Although results with the smaller datasets are somewhat surprising, MCMC does a good job with Toy Retail and $p = 1000$. It only misses the weak arc from $P$ to $S$. It is also one of the two algorithms (together with GES, see Subsection 4.4.1) to output a correct minimal I-map for the XOR problem.

### 4.2.4 Search-and-Score Algorithms with Incomplete Datasets

Incomplete datasets are often what we will find in real-world situations. Missing values are commonplace. A way of getting rid of them is simply ignoring the data points containing missing

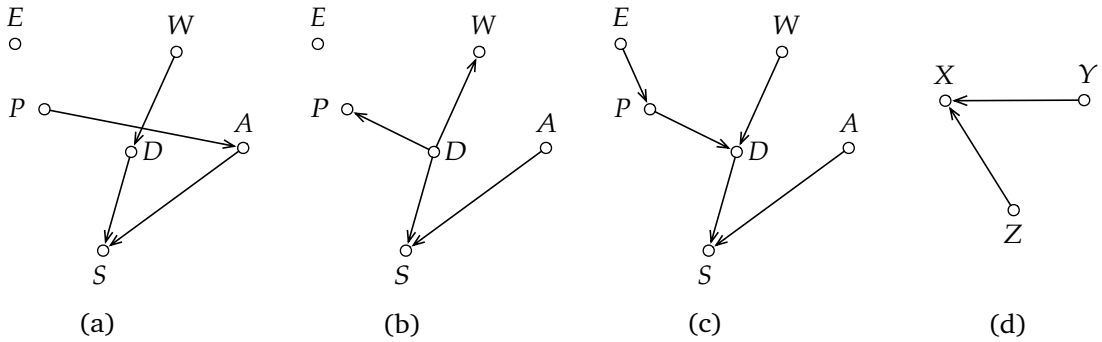



FIG. 4.7: Graphs for MCMC: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

values, but this can induce a bias in the results if the values are not missing at random but are dependent on some other observed or unobserved variables.

In general, we distinguish three cases of missing values:

- **Missing Completely At Random (MCAR)**

  The probability for a value to be missing does not depend on observed or unobserved data. In this case, constructing the network having deleted the data points with missing values will not produce biased results, although information might be lost.

  However, in some datasets with a large number of features, the probability of still having complete data points might become small, such that the resulting dataset could be considerably smaller than the original one. In the MCAR case, consistent estimators for the missing data can be found.

- **Missing At Random (MAR)**

  The probability for a value to be missing only depends on the observed data. Dependencies have to be found out and consistent estimators can be found if conditioned on the dependent variables.

- **Not Missing At Random (NMAR)**

  Missingness depends both on observed and unobserved data. Unless we also build a model of how data is missing with additional expert knowledge, we cannot build consistent estimators from the observed data and thus cannot complete it. The results will be biased, whether we use estimators based on the observed data or we ignore the uncomplete data points.

We cannot tell whether a dataset with missing values is MCAR, MAR or NMAR generally, but we can differentiate between MCAR and MAR by looking for dependencies in the observed data, once we assume that there are no dependencies on the unobserved data. See Leray & François (2005), Riggelsen (2006) or the website by Carpenter & Kenward (2006) and their large bibliography for more detailed definitions and discussions.

The presented techniques here work with the MCAR and MAR missingness types.

### Structural EM (S-EM)

Structural EM (Friedman, 1997, 1998) combines the techniques of greedy search and of missing value estimation using an expectation-maximization (EM) algorithm with MAP estimators. Structural EM performs a search in the joint space of parameters and structure. At each step, it either finds new estimates of the missing values by updating the parameters or switches to a better DAG structure. It requires more time to terminate than any of the previously described algorithms.

An EM algorithm, like the greedy search we described, is a hill-climbing technique which converges to a local optimum. Structural EM works with whole Bayesian networks rather than just with DAGs because it needs to make inference to estimate the missing parameters.

In order to benchmark this algorithm, we have run it the same dataset three times, removing successively $m = 10\%$, $25\%$ and $50\%$ percent of the values randomly.

The constructed graphs are shown in Figure 4.8 and run times in Table 4.1. The graphs for the XOR problem were omitted because they were all empty.

| Missing | Toy Retail $p = 50$ | Toy Retail $p = 200$ | Toy Retail $p = 1000$ | XOR $p = 1000$ |
|---|---|---|---|---|
| $m = 10\%$ | 48 s | 3 min 3 s | 21 min 39 s | 4 min 22 s |
| $m = 25\%$ | 14 s | 3 min 4 s | 16 min 24 s | 5 min 33 s |
| $m = 50\%$ | 24 s | 1 min 37 s | 4 min 50 s | 8 min 55 s |

TABLE 4.1: Run times of the Structural EM algorithms with the test datasets

Considering the long run times and the resulting graphs, we see that the problem of taking into account missing values is tough and that today's approaches lack efficiency. It would probably be worth conducting more tests with larger datasets to check the convergence of the algorithm. The high missing data probabilities were chosen for the sake of demonstration and do not necessarily reflect what can typically be found in real-world scenarios.

### Generic EM

The number of publications presenting approaches which deal with missing data is smaller than the number of publications dealing with complete datasets. Looking at the previous algorithm, we can extract a generic EM wrapper structure to deal with incomplete datasets of the type MAR or MCAR:

1. Start with some (good) initial structure;
2. Run the algorithm with the data rows that are complete (or have been completed by a previous iteration). This could be either an iteration of the algorithm or the whole algorithm if it cannot be easily decomposed into steps where each of them would provide a valid structure;
3. Find estimates of the missing data using the obtained structure and update the dataset with them;
4. Go to Step 2 until no further improvement can be obtained.

Of course, convergence does not come for free and has to be ensured. Note that finding estimates of missing values usually requires that the temporary Bayesian/causal network be fully specified, i.e., that its parameters be also estimated.

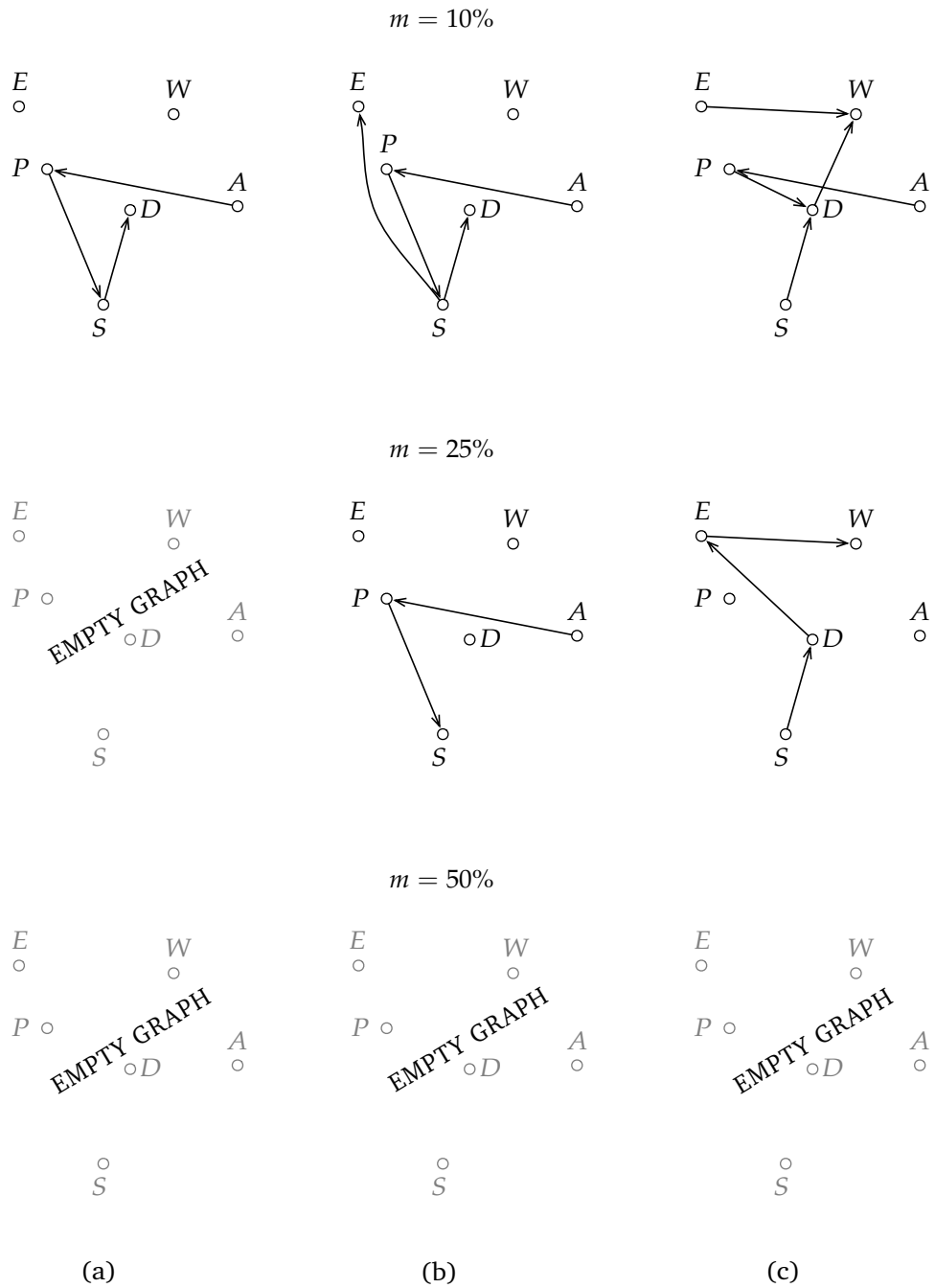



FIG. 4.8: Graphs for S-EM: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$

#### Other Missing Data Approaches

Other approaches to structure learning with missing data have been proposed. The methods described in the following references have not been tested or implemented for this study. Myers *et al.* (1999) describe a stochastic search approach with mutation operators in order to avoid getting stuck in local optima. Leray & François (2005) propose an EM algorithm working in the space of trees rather than DAGs. Other papers discussing the missing values problem include Thiesson (1995); Singh (1997); Meila & Jordan (1998).

## 4.3 Learning Causal Networks: Assumptions

In order to attach a causal meaning to the arcs of a causal graph based on observational data, we need extra assumptions, without which the validity of the proposed cause–effect relationships cannot be asserted.

### 4.3.1 Causal Markov Condition

Formally, the causal Markov condition states that in a causal graph $\mathcal{G}$, for all distinct variables $X, Y \in \mathbf{V}$, if $X$ is not a direct or indirect cause for $Y$, then $p(Y \mid X, \mathbf{Pa}(X)) = p(Y \mid \mathbf{Pa}(X))$. Said differently: if $Y \in \mathbf{Nd}(X)$, it is shielded from $X$ by the parents of $X$ and an intervention on $X$ will have no effect on $Y$.

A discussion of the causal Markov condition and equivalent definitions can be found in Hausman & Woodward (1999). This condition can be interpreted as making the link between conditional dependence and causation, and implies that if $X$ and $Y$ are dependent conditional on the set of all directs parents of $X$, then $X$ causes $Y$ (directly or indirectly), i.e., $Y \in \mathbf{De}(X)$.

Note that this condition alone can be trivially fulfilled by any distribution where each variable is independent of all others. This is to be related to the D-map property.

### 4.3.2 Faithfulness Condition

The Faithfulness condition can be seen as the converse of the causal Markov condition and is thus to be related to the I-map property. It says that the only independencies to hold in the data are those determined graphically by the causal graph and the causal Markov condition. More on the Faithfulness condition can be found in Steel (2005).

An implication of this condition is that all independencies of the data must be structural and not parametric, a concept which Pearl (2000) refers to as *stability*. It is illustrated in Figure 4.9. In this graph, no independency is represented. However, it is possible through careful parametrization to have the effect of $X$ on $Y$ cancelled out by the chained effect of $X$ on $Z$ and of $Z$ on $Y$. $X$ would then be independent of $Y$, although the structure of the graph says otherwise. In a linear SEM, this happens if the path coefficients satisfy $\alpha = -\beta\gamma$. Faithfulness ensures that we avoid such "lucky distributions."

Together, the causal Markov condition and the faithfulness condition entail a one-to-one mapping between the conditional independencies and the $d$-separation criterion. Remembering the P-map

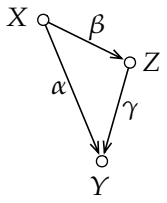



FIG. 4.9: An sample graph where a stability problem may occur

property, this requires that the underlying set of conditional independencies be DAG-isomorphic. So, whereas these conditions are often introduced as assumptions in the literature, we prefer the formulation saying that the actual assumption is *DAG-isomorphicity* of the probability distribution represented by our data, and that causal Markov and Faithfulness are then really two conditions that can be met thanks to this assumption.

### 4.3.3 Causal Sufficiency Assumption

The Causal Sufficiency assumption says that the observed variables $\mathbf{V}$ include all common causes for pairs of variables in $\mathbf{V}$.

This important assumption protects us from latent variables and spurious associations. This is maybe the most controversial assumption about causal inference because it is the one which guarantees that a detected arc between two variables is indeed an unmediated effect and not a side effect due to a latent common cause.

In real life, this assumption can rarely be verified. Suppose we conduct an experiment to determine whether smoking causes lung cancer and we find a strong positive correlation between the two variables. Assuming Causal Sufficiency, two causal graphs are plausible:

$$\text{SMOKING} \longrightarrow \text{LUNG CANCER}$$
$$\textit{or} \quad \text{LUNG CANCER} \longrightarrow \text{SMOKING}.$$

Looking at temporal information, we decide to rule out the second possibility and conclude that smoking does cause lung cancer.

If, however, the Causal Sufficiency assumption is contested by a third party, then a further structure must be taken into account, integrating some latent common cause $L$ to get this causal graph:

$$\text{SMOKING} \longleftarrow L \longrightarrow \text{LUNG CANCER}.$$

We could thus refute the conclusion that smoking causes lung cancer.

It is therefore a strong assumption, which is related to the causal underidentification. Some algorithms do not assume Causal Sufficiency and can then hint at the presence of latent variables, or, in certain circumstances, determine that a link is probably a true direct cause–effect relationship. These inferences are all based on the V-structure detection principle (cf. Subsection 2.3.2) to introduce or exclude converging nodes, and require at least four variables to have a chance to determine the presence of a latent variable or of a genuine causal effect between two of them.

## 4.4 Learning Causal Networks: Algorithms

We now turn to algorithms which are suitable for learning the structure of causal networks. Because causal networks can be defined as special cases of Bayesian networks with a causal meaning attached to the arcs, the following algorithms will also learn valid Bayesian networks.

Remembering the causal underidentification problem, we must now find a way to signal when some causal relationship cannot be directed. This is done with the help of partially directed acyclic graphs (PDAGs) and ancestral graphs (AGs). All of our causal structure learning algorithms return completed PDAGs (CPDAGs), which are DAGs with some undirected edges but maximally oriented with respect to the conditional independencies, or AGs, which, roughly, are PDAGs with bidirected edges, or some other variants.

The meaning of a directed arc is clear: it depicts a direct cause–effect relationship. An undirected edge represents an unorientable causal relationship, and a bidirected edge usually indicates the presence of a common, hidden cause between two variables.

In order to detail these algorithms a bit more, pseudocode is also listed for each of them in Appendix A.

### 4.4.1 Greedy Equivalence Search (GES)

Greedy Equivalence Search (Meek, 1997; Chickering, 2002) is a variant of the greedy search proposed before but in the space of the independence equivalent (Markov equivalent) DAGs, i.e., of the corresponding PDAGs. It keeps only adding edges until the score cannot be improved any more, then considers removing extraneous edges.

This algorithm has been proved to converge to the true P-map of the data, provided its conditional independencies are DAG-isomorphic, and does not get stuck in local optima, despite its greedy approach. Like GS, the GES algorithm calls the scoring functions heavily and is computationally expensive. It belongs to one of the state-of-the-art structure construction algorithms today. Pseudocode can be found in Algorithm 2 in Appendix A.

The constructed graphs are shown in Figure 4.10. Run times were 3.83 s, 4.79 s, 6.31 s and 0.95 s, respectively.

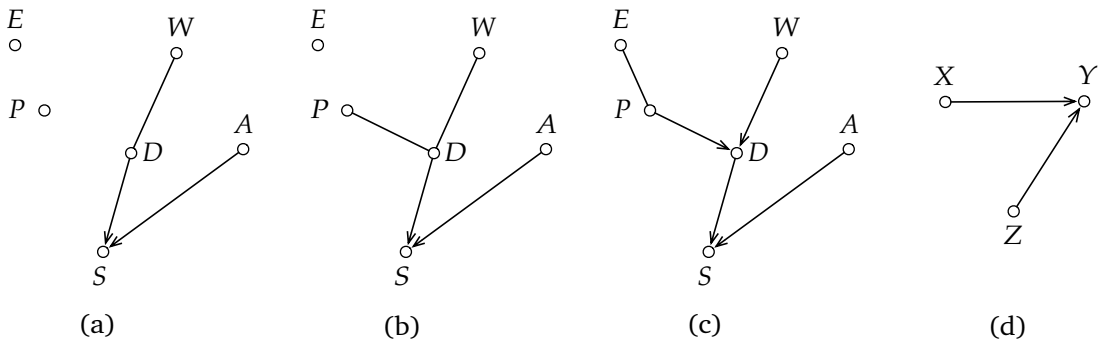



FIG. 4.10: Graphs for GES: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

Except for the weak $P \rightarrow S$ feature, GES detects only correct links and the accuracy of the obtained graphs increases with the size of the dataset. The case $p = 1000$ allows GES to output the CPDAG of the equivalence class of the problem structure (up to the missing weak link). Interestingly, it does also output an I-map for the XOR test, although its DAG-constructing counterpart GS returns an empty graph.

### 4.4.2 Inductive Causation (IC/PC)

The IC algorithm, introduced in Pearl & Verma (1991), and its more detailed version but otherwise equivalent, the PC algorithm,[2] introduced in Spirtes *et al.* (1993), do not use scoring functions, but perform conditional independence tests and build the structure according to the constraints represented by the outcome of these tests. The algorithms can be outlined in three phases:

1. Find the undirected structure: add an undirected edge between $X$ and $Y$ if no set $\mathbf{S}_{XY}$ can be found such that $(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$.
2. Determine the V-structures. For each connected triple $X - Z - Y$, direct the edges and add a V-structure $X \rightarrow Z \leftarrow Y$ if $Z \notin \mathbf{S}_{XY}$, i.e., if and only if $X$ and $Y$ are *dependent* given $Z$.
3. Direct the remaining edges. Follow different rules/constraints to direct the remaining edges whenever possible, avoiding the creation of new V-structures and of cycles.

The first step (looking for $\mathbf{S}_{XY}$ for all pairs) can be done in polynomial time $\mathcal{O}(n^k)$ given a maximum graph fan-in $k$. The algorithm looks for independence between each pair of variables by starting with sets of cardinality 0, then cardinality 1, and so on, removing edges from a complete graph as soon as separation is found for a given cardinality. The search for a separating set $\mathbf{S}_{XY}$ can then be limited to nodes adjacent to $X$ and $Y$. The PC algorithm is detailed in Algorithm 3 in Appendix A.

The only edges which can be directly directed are the ones which are part of V-structures. The additional constraints we can propagate come from the observation that additional edge orientations may not create cycles nor additional V-structures. The question arises to know whether the proposed rules are enough to maximally orient the graph without arbitrarily orienting some edges. Pearl & Verma (1991) originally proposed four rules, but then Meek (1995) proved that three of them, described in the pseudocode, are enough to obtain the CPDAG described by the set of conditional independencies.

The constructed graphs are shown in Figure 4.11. Run times were 0.79 s, 0.40 s, 1.91 s and 0.09 s, respectively. The higher run time for the smallest problem comes from warnings issues by Matlab on the lack of significance of the conditional independence tests.

The algorithms working with conditional independence tests suffer from the lack of sufficiently large datasets (this issue is further discussed in Subsection 4.6.3). These tests fail to detect the independencies with a small number of samples and this typically includes too many arcs, as the results of IC/PC, IC* and CBL show. With more samples, however, IC/PC does recover the same PDAG as GES.

It fails on the XOR test, because each link between any two variables is discarded, having an unconditional independency between them.

[2] "PC" stands for "Peter" and "Clark," the first names of the authors of the algorithm.

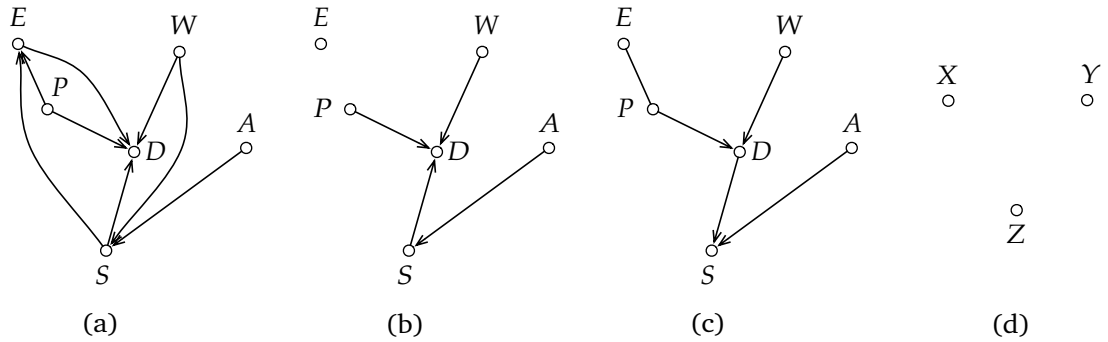



FIG. 4.11: Graphs for IC/PC: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

### 4.4.3 Inductive Causation with Latent Structures (IC*)

A modification of the IC/PC algorithm allows to look for latent structures, too; i.e., for potential hidden variables. We are thus relaxing the Causal Sufficiency assumption, but we replace it with the assumption that all hidden variables to be found have each two children and are independent from one another. The output of the algorithm is then an ancestral graph with some marked edges ($\stackrel{*}{\rightarrow}$), called a *latent structure* in Pearl (2000). The edges found in the resulting graph are now one of the following:

- $X \stackrel{*}{\rightarrow} Y$ (marked) indicates a directed arrow $X \rightarrow Y$ in the DAG projection (a *genuine* cause).
- $X \rightarrow Y$ (normal) indicates either $X \rightarrow Y$ or a latent common cause $X \leftarrow L \rightarrow Y$.
- $X \leftrightarrow Y$ (bidirected) indicates a latent common cause $X \leftarrow L \rightarrow Y$.
- $X - Y$ (undirected) indicates either $X \rightarrow Y$, $Y \rightarrow X$ or a latent cause $X \leftarrow L \rightarrow Y$.

Step 2 of the IC/PC algorithm is modified in the sense that when we detect a V-structure in the triplet $X - Z - Y$, we add arrowheads into $Z$ instead of definitely orienting the edges. This allows to create bidirected arrows. The relevance of this modification can be illustrated by the following example.

Suppose we have the following simple undirected structure: $X - Z - W - Y$ and have $Z \in \mathbf{S}_{XW}$ and $W \in \mathbf{S}_{ZY}$ after Step 1. The first statement can be interpreted as $X$ and $W$ being independent, while conditioning on $Z$ creates the dependency. This is typically the V-structure case, so we orient our graph as $X \rightarrow Z \leftarrow W - Y$ to find the only structure to be compatible with our first statement. Now, looking at the second statement, the same reasoning can be applied, to find that the only compatible structure is $X - Z \rightarrow W \leftarrow Y$. Combining the two structures, we have $X \rightarrow Z \leftrightarrow W \leftarrow Y$, but the bidirected arrow represents something we are not familiar with in terms of causality. The easiest interpretation consists of adding a latent common cause $X \rightarrow Z \leftarrow L \rightarrow W \leftarrow Y$, which is the simplest modified DAG consistent with our initial undirected structure and statements.

Step 3 of the IC/PC algorithm is replaced by the following. The arrow $X \ast\!\!\rightarrow Y$ means any directed, bidirected, marked arc that has an arrow pointing into $Y$; similarly, $X \ast\!\!- Y$ means any edge that does *not* have an arrow pointing into $Y$.

3. Direct and mark as many edges as possible according to the following rules:

(a) $X \ast\!\!\rightarrow Z - Y$ becomes $X \ast\!\!\rightarrow Z \overset{*}{\rightarrow} Y$;
(b) $X \ast\!\!- Y$ becomes $X \ast\!\!\rightarrow Y$ if there is a marked directed path from $X$ to $Y$.

Note that the new rules are also working on arrowheads rather than on edges, which allows an undirected edge to become bidirected.

Do these rules make sense? Rule (a) prevents new V-structures from being created by introducing marked arcs. According to the new definition of edges, these marked arcs are indeed the only way which prevents the introduction of a latent variable as common cause, and the latent variable scenario is excluded in the context of Rule (a) because otherwise a V-structure would have been detected in Step 2 with $Z$ as converging node. Rule (b) simply prevents the introduction of cycles—we now require the potentially existing path between $X$ and $Y$ to be fully marked, because it could otherwise include a latent common cause between two nodes, which would break the cycle.

The pseudocode for the IC* algorithm can be found in Algorithm 4 in Appendix A.

Before looking at results with this algorithm, it is worth looking closer at the distinction that is made by the algorithm between potential causation ($\rightarrow$), genuine causation ($\overset{*}{\rightarrow}$), and spurious association ($\leftrightarrow$) as described in Pearl (2000).

**Definition 4.4.1 (Potential Cause)** *A variable $X$ is a* potential cause *for $Y$ if and only if:*

1. *$X$ and $Y$ are unconditionally dependent;*
2. *There exist a variable $Z$ and a set of variables $\mathbf{S}$ such that $(X \perp\!\!\!\perp Z \mid \mathbf{S})$ and $(Y \not\perp\!\!\!\perp Z \mid \mathbf{S})$.*

$Z$ thus represents a variable which also influences $Y$ (but not $X$) given a certain context. Typical examples are V-structures.

**Definition 4.4.2 (Genuine Cause)** *A variable $X$ is a* genuine cause *for $Y$ if and only if:*

1. *$X$ and $Y$ are unconditionally dependent;*
2. *There exist a variable $Z$ and a set of variables $\mathbf{S}$ such that $Z$ is a potential cause for $X$, $(Y \not\perp\!\!\!\perp Z \mid \mathbf{S})$ and $(Y \perp\!\!\!\perp Z \mid \mathbf{S} \cup \{X\})$,*

or: *$X$ and $Y$ are in the transitive closure of the relation defined by Points 1 and 2.*

By the conditions listed in Rule 2, we make sure that there exists no path from a parent of $X$ to $Y$ that does not go through $X$ itself.

**Definition 4.4.3 (Spurious Association)** *Two variables $X$ and $Y$ are* spuriously associated *if and only if there exist variables $Z_1$, $Z_2$ and contexts $\mathbf{S_1}$, $\mathbf{S_2}$ such that:*

1. *$(Y \not\perp\!\!\!\perp Z_1 \mid \mathbf{S_1})$ and $(X \perp\!\!\!\perp Z_1 \mid \mathbf{S_1})$;*
2. *$(X \not\perp\!\!\!\perp Z_2 \mid \mathbf{S_2})$ and $(Y \perp\!\!\!\perp Z_2 \mid \mathbf{S_2})$.*

Condition 1 disqualifies $Y$ as a cause for $X$ and Condition 2 disqualifies $X$ as a cause for $Y$. The only remaining explanation is a hidden common cause for $X$ and $Y$.

We now turn to the output of IC* when given our test data, shown in Figure 4.12. Run times were 0.35 s, 0.37 s, 1.61 s and 0.16 s, respectively.

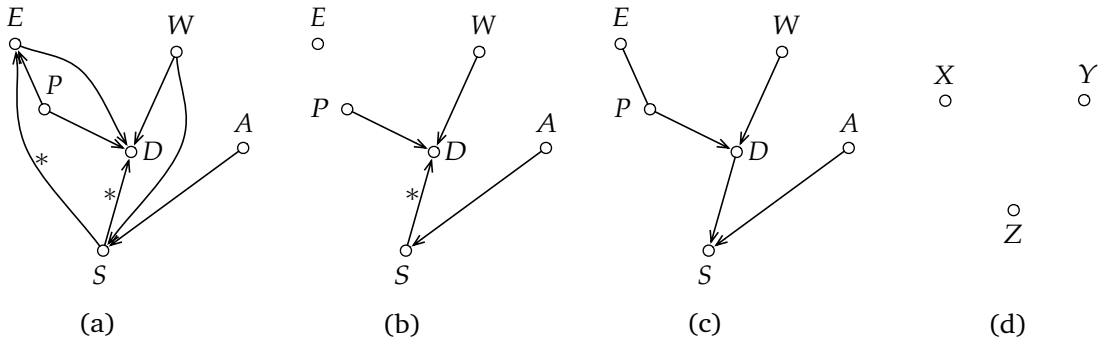



FIG. 4.12: Graphs for IC*: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

Not surprisingly, IC* yields the same DAG projections of the results as IC/PC, as only the orientation constraint propagation step changes. It correctly omits to detect latents variables, but the only arcs it marks as being true causation are either extraneous or misdirected. Note that with $p = 1000$, IC* asserts that all arcs denote potential and not genuine causation.

Another similar algorithm to construct a causal structure with detection of latent variables has been proposed in Spirtes *et al.* (1995) but was not tested in this study.

### 4.4.4 Cheng-Bell-Liu Algorithm (CBL)

This algorithm (Cheng & Bell, 1997; Cheng *et al.*, 1997a) always runs in polynomial time. If a prior ordering is given, it performs $\mathcal{O}(n^2)$ conditional independence tests; if it has to orient the edges itself, it performs $\mathcal{O}(n^4)$ tests (Cheng *et al.*, 1997b). It works in three or four phases, depending on whether arcs must be oriented by the algorithm.

1. *Drafting.* Create edges according to the pairwise mutual information of the variables (or other similarity measure), making sure we do not create new paths between nodes that are already $d$-connected. If a prior ordering is available, edges are oriented according to it, else, the structure remains temporarily undirected. Put node pairs whose mutual information is not negligible but whose corresponding edge would create additional paths between the two nodes into a list **R**.

2. *Thickening.* Examine each remaining variable pair in **R** (i.e., each remaining variable pair whose mutual information is not negligible) to find with a conditional independence test if these variables are still dependent given the minimum $d$-separating set (extracted from the current graph). If they are, add a new edge between them. This step ensures that the graph becomes an I-map with respect to the given dataset.

3. *Thinning.* For each edge $X - Y$: remove the edge if a conditional independence test shows that $X$ and $Y$ are not dependent given the minimum $d$-separating set between $X$ and $Y$ that can be found in the graph where the edge $X - Y$ as temporarily been removed.

4. *Orienting.* If no prior ordering is given, this phase orients as many edges as it can, following a procedure that includes additional conditional independence tests.

Note that $d$-separation tests are not feasible if no prior ordering is given, as the graph being built is undirected. These tests and the procedure for finding minimum $d$-separating sets are replaced by other (similar) heuristics.

Also note that Phase 1 is closely related to the MWST algorithm in the sense that it also builds a tree with respect to some similarity measure. Pseudocode in given in Algorithm 5 in Appendix A.

The constructed graphs are shown in Figure 4.13. Run times were 1.91 s, 0.49 s, 0.27 s and 0.35 s, respectively. The surprisingly decreasing run times are due the warning messages output by Matlab with the smaller datasets.

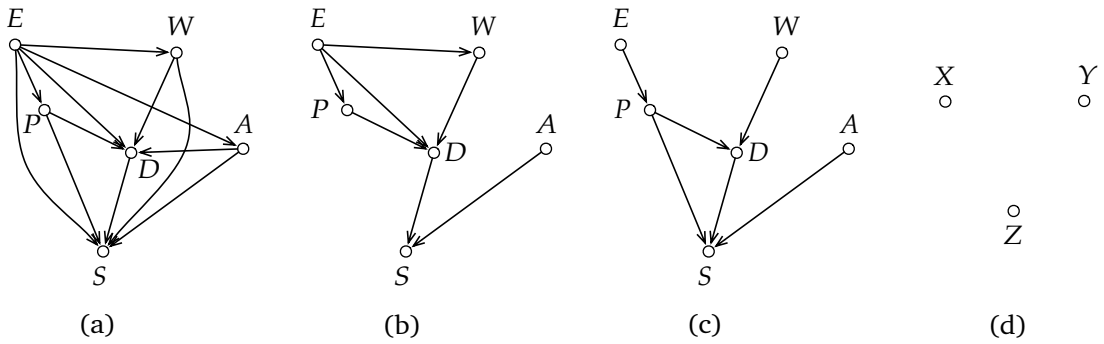



FIG. 4.13: Graphs for CBL: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

Although CBL detects too many links with small datasets, it is the only algorithm to output a graph with no missing arc with $p = 1000$ (note that the implemented version of CBL is the one using a prior ordering to orient the arcs). But, using only a pairwise similarity measure, it also fails at the XOR test.

## 4.5 COMPARISON OF ALGORITHMS

We now sum up the reviewed algorithm in Table 4.2. The abbreviations in the Type column correspond either to "Search-and-Score" (S&S) or to "Construction under constraints" (CC). The numbers in parentheses refer to the notes below.

There is no algorithm for dealing with missing values while constructing the network under constraints. If missing values are of type MCAR, it is possible to remove the corresponding data points at the expense of information loss. It is also possible to ignore data points only when considering features which are missing for the considered task. For example, suppose we have $\mathbf{V} = \{X, Y, Z, W\}$, and the value for $W$ is missing in data point $j$. Then, obviously, we need not remove this data point from the considered dataset when determining the similarity between $Y$ and $Z$ or whether $(X \perp\!\!\!\perp Y \mid Z)$ holds, but we will skip it if we want to know $(X \perp\!\!\!\perp W \mid Z)$, for instance.

The continuous variable case is a problem: only MWST supports it. As this algorithm is usually intended to bootstrap other algorithms, the fact that it handles continuous variables is, in our case, limited. Notably, none of our causal algorithms explicitly deals with them. As Note 2 says, this is further discussed in the next section and chapter.

| Algorithm | Type | Missing values? | Continuous variables? | Latent variables? | Work w/o ordering? | Output |
|---|---|---|---|---|---|---|
| K2 | S&S | — | — | — | — | DAG |
| MWST | S&S | — | √ | — | √ (1) | Tree |
| GS | S&S | — | — | — | √ | DAG |
| MCMC | S&S | — | — | — | √ | DAG |
| S-EM | S&S | √ | — | — | √ | DAG |
| GES | S&S | — | — | — | √ | CPDAG |
| IC/PC | CC | — | — (2) | — | √ | CPDAG |
| IC* | CC | — | — (2) | √ | √ | AG (3) |
| CBL | CC | — | — (2) | — | — | CPDAG |
| CBL 2 | CC | — | — (2) | — | √ | CPDAG |

TABLE 4.2: Comparison of structure construction algorithms: typology

1. No ordering is required, but a root node for the tree has to be designated.
2. The compatibility with continuous variables depends on finding a good way to evaluate similarity measures and conditional independence tests for continuous variables; this issue is discussed in the next section.
3. The returned AG may also also contain marked arcs.

The search-and-score methods presented here do not look for possible latent variables. It indeed blows up the search space even more. Elidan *et al.* (2000) discuss the issue of integrating latent variable detection into a structure search algorithm. Scheines *et al.* (1995) and Silva *et al.* (2006) discuss the identification of latent variables with the help of the so-called tetrad difference.

Recapitulating the runs of the different algorithms and their errors, we show in each cell of Table 4.3 an entry of the form "$x$, $y$ ($z$)" where $x$ is the number of missing arcs, $y$ the number of extraneous links, and $z$ the number of wrongly directed arcs. For the XOR problem, we simply state whether the recovered structure was an I-map of the problem. Note that K2 and the version of CBL we implemented require a prior ordering and thus make no orientation mistakes if the ordering is correct.

| Algorithm | Toy Retail $p = 50$ | Toy Retail $p = 200$ | Toy Retail $p = 1000$ | XOR $p = 1000$ |
|---|---|---|---|---|
| K2 | 3, 0 (0) | 2, 0 (0) | 1, 0 (0) | — |
| MWST | 1, 0 (2) | 1, 0 (2) | 1, 0 (2) | — |
| GS | 3, 0 (2) | 1, 0 (3) | 1, 2 (5) | — |
| MCMC | 3, 1 (0) | 2, 0 (2) | 1, 0 (0) | √ |
| GES | 3, 0 (0) | 2, 0 (0) | 1, 0 (0) | √ |
| IC/PC | 1, 3 (2) | 2, 0 (1) | 1, 0 (0) | — |
| IC* | 1, 3 (2) | 2, 0 (1) | 1, 0 (0) | — |
| CBL | 0, 6 (0) | 1, 2 (0) | 0, 0 (0) | — |
| S-EM: | | | | |
| $m = 10\%$ | 4, 1 (1) | 4, 2 (1) | 3, 2 (2) | — |
| $m = 25\%$ | 6, 0 (0) | 5, 1 (0) | 5, 2 (1) | — |
| $m = 50\%$ | 6, 0 (0) | 6, 0 (0) | 6, 0 (0) | — |

TABLE 4.3: Comparison of structure construction algorithms: learning errors

While results with small datasets differ significantly, the graph seems to stabilize with a growing number of samples for all algorithms but GS. The causal discovery algorithms output the

same CPDAG, correctly detecting the V-structures, and CBL even recovers the weakest association. MCMC correctly directs all detected arcs, including the one which is undirected in the equivalence class. But violation of the DAG-isomorphicity assumption is a big problem, even for a very simple structure. Only MCMC and GES can recover an I-map of the XOR problem; in particular, algorithms constructing the network with conditional independence tests systematically return an empty graphs. Structural EM is a bit off-target in the context of our tests and has a high error rate.

## 4.6 IMPLEMENTATION DETAILS

Several algorithms use similarity measures, conditional independence tests, the $d$-separation criterion and minimal $d$-separating sets to build the network structure. This is not always trivial to implement and a few techniques are detailed here.

### 4.6.1 Similarity Measures

We have several options regarding how to implement a similarity measure for a variable pair. First, let us quickly review what a *distance* is in the mathematical sense (Widdows, 2004): a distance is a function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ defined on a given set $\mathcal{M}$ which meets the following criteria:

- Positiveness: $d(x,y) \geq 0$, and $d(x,y) = 0$ if and only if $x = y$;
- Symmetry: $d(x,y) = d(y,x)$;
- Triangle Inequality: $d(x,z) \leq d(x,y) + d(y,z)$ (the shortest distance between two points is on a straight line).

Let's now review some (dis)similarity measures (which can be distances or not).

- **Euclidean Distance**

  The simple Euclidean distance between the two vectors/series representing the outcome of the random variables. Only usable to detect linear dependencies; usually requires the data to be normalized or to be known to have very similar ranges. In our case, this technique is not adequate.

- **Pearson's Correlation Coefficient**

  The correlation coefficient between two random variables $X$ and $Y$ is:

  $$\rho_{XY} = \frac{\text{cov}(X,Y)}{\sqrt{\text{var}(X)}\sqrt{\text{var}(Y)}} = \frac{\mathbb{E}\{[X - \mathbb{E}(X)][Y - \mathbb{E}(Y)]\}}{\sqrt{\mathbb{E}\{[X - \mathbb{E}(X)]^2\}}\sqrt{\mathbb{E}\{[Y - \mathbb{E}(Y)]^2\}}}. \tag{4.17}$$

  This correlation is only able to detect linear dependencies between the variables. Actually, it is possible to find cases where a variable is fully dependent on another but where their correlation is close to zero. It is now an open question to assess how appropriate the correlation will be in our problem.

- **Mutual Information**

  The mutual information between two variables $X$ and $Y$ is given as:

  $$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}. \tag{4.18}$$

The continuous case becomes:

$$I(X;Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \, dxdy. \tag{4.19}$$

Contrary to Pearson's correlation, the mutual information is sensitive to any functional relationship, and is zero if and only if $X$ and $Y$ are independent. The mutual information is symmetrical but does not satisfy the triangle inequality, and is therefore not a distance. A mathematical distance based on mutual information can be defined as $D(X,Y) = H(X,Y) - I(X;Y)$. See Steuer *et al.* (2002).

- **Kullback-Leibler Divergence**

  Another measure drawn from information theory. KL divergence is asymmetrical. It is defined as follows (discrete and continuous cases):

$$D_{\mathrm{KL}}(P\|Q) = \sum_{k \in \mathcal{K}} P(k) \log \frac{P(k)}{Q(k)} \tag{4.20}$$

$$D_{\mathrm{KL}}(P\|Q) = \int_{\mathcal{K}} p(k) \log \frac{p(k)}{q(k)} \, dk \tag{4.21}$$

  The KL divergence can be interpreted as the information gain when replacing the initial distribution $Q$ by $P$. It is zero when $P$ and $Q$ are identical and is otherwise always positive. It can be used as similarity measure when two variables with the same range $\mathcal{K}$ are to be compared.

Our choice will often be either Pearson's correlation or mutual information—keeping in mind the specific features of each. Correlation is easy to compute for continuous variables, but does not detect any kind of functional relationship, and it is not well-suited for categorical attributes. Correlation between two variables can be computed in linear time with respect to the sample size ($\mathcal{O}(p)$). Mutual information is easy to compute for categorical/discrete variables (also $\mathcal{O}(p)$), but requires more computationally expensive techniques for continuous variables, a few of which we review now.

### 4.6.2 Mutual Information for Continuous Variables

The problem with the integral in Equation 4.19 describing the mutual information between two continuous random variables is that we do not know the density functions $p(x)$, $p(y)$ and $p(x,y)$ but only have samples (which we assume are i.i.d.) drawn from them. To compute the integral, we can either discretize the data, partitioning it into bins and use the sum in Equation 4.18, or we use density estimation techniques to find estimates $\hat{p}(x)$, $\hat{p}(y)$ and $\hat{p}(x,y)$ for the density functions.

#### Histogram techniques

Data is partitioned into $p_X$ equally-sized bins for $X$ and $p_Y$ bins for $Y$. Then, with a sample size $p$, $\hat{p}(x_i) = k_i/p$ and our approximation is

$$\hat{I}(X;Y) = \log p + \frac{1}{p} \sum_{i=1}^{p_X} \sum_{j=1}^{p_Y} k_{ij} \log \frac{k_{ij}}{k_i k_j}. \tag{4.22}$$

Steuer *et al.* (2002) showed that

$$\mathbb{E}_{X,Y}[\hat{I}(X;Y)] \approx I(X;Y) + Err_I \tag{4.23}$$

with

$$Err_I = \frac{p^*_{XY} - p^*_X - p^*_Y + 1}{2p}, \tag{4.24}$$

where $p^*_X$, $p^*_Y$ and $p^*_{XY}$ are the number of discrete states with a nonzero probability.

**Kernel Density Estimation**

Detailed information on this method can be found in Moon *et al.* (1995). The idea here is twofold: first, free the histogram from a fixed origin determined by the size of the bins in the previous approach. For instance, we could estimate the density at $x$ by counting the number of points in a neighborhood of width $2h$ ($\Theta(\cdot)$ is the heaviside function):

$$\hat{p}(x) = \frac{1}{2ph} \sum_{i=1}^{p} \Theta(h - |x - x_i|). \tag{4.25}$$

Second, free the histogram from its rectangular shape. More generally, we can replace the rectangular bins by a kernel $k(\cdot)$:

$$\hat{p}(x) = \frac{1}{ph} \sum_{i=1}^{p} k\left(\frac{|x - x_i|}{h}\right), \tag{4.26}$$

for instance, a Gaussian kernel:

$$\hat{p}(x) = \frac{1}{ph} \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{p} e^{-\frac{|x-x_i|^2}{2h^2}}. \tag{4.27}$$

Special care must be taken to choose $h$. According to Silverman (1986), we have, if the density being estimated is Gaussian-distributed, $h_{opt} = \sqrt[5]{4/(3p)} \cdot \hat{\sigma}$, where $\hat{\sigma}$ is the sample standard deviation of the data.

The joint probability can be estimated similarly:

$$\hat{p}(x,y) = \frac{1}{ph^2} \frac{1}{2\pi} \sum_{i=1}^{p} e^{-\frac{|x-x_i|^2 + |y-y_i|^2}{2h^2}}. \tag{4.28}$$

We can now numerically integrate the integral describing the mutual information using standard numerical methods. Alternatively, a simpler way to do it is to consider that we can write

$$\hat{I}(X;Y) = \mathbb{E}_{X,Y}\left(\log \frac{\hat{p}(x,y)}{\hat{p}(x)\hat{p}(y)}\right). \tag{4.29}$$

Under the assumption that the dataset is a representative sample of the underlying distribution,

we obtain this empirical estimation:

$$\hat{I}(X;Y) = \frac{1}{p}\sum_{i=1}^{p} \log \frac{\hat{p}(x_i, y_i)}{\hat{p}(x_i)\hat{p}(y_i)}. \tag{4.30}$$

The trick which lets us leave out the multiplying probability from the original definition of the mutual information is that we are now summing over the data points instead of over the whole space $\mathcal{X} \times \mathcal{Y}$.

This algorithm has complexity $\mathcal{O}(p^2)$.

Evaluating $p$ times the probability density function with kernel estimators is inefficient because of its quadratic complexity. Methods have been developed to efficiently compute sums of Gaussians, like the Fast Gauss Transform (Greengard & Strain, 1991; Elgammal *et al.*, 2003; Yang *et al.*, 2003). A dedicated Matlab toolbox (Ihler, 2004) is available and was used in our implementation.

### 4.6.3 Conditional Independence Tests

Traditional (unconditional) independence tests are for instance the $\chi^2$ test for independence or the likelihood ratio test. Checking for vanishing mutual information (or vanishing correlation if our assumptions allow to equate a vanishing correlation with independence) is also an alternative.

Most of the time, however, our goal is to perform *conditional* independence tests. The conditioning set can have any size (actually up to $n-2$, because we will never condition on one of the two variables between which we are measuring the similarity).

Suppose we want to know whether $X$ and $Y$ are conditionally independent, and the conditioning set has cardinality 1 and contains a multinomial variable $Z$ with $k$ possible values. Then we can perform $k$ independence tests between $X$ and $Y$ drawn from the $k$ datasets $D_i$, where $D_i$, $1 \leq i \leq k$, is the dataset containing only the data points where $Z = z_i$. Conditional independence is then ensured when all independence tests succeed. Note that the average amount of data to perform the unconditional test on is reduced by a factor $k$. Suppose now that the conditioning set has cardinality 2 with two $k$-valued variables: the resulting datasets will then be smaller by a factor $k^2$. To keep the power of the conditional test constant, the size of the dataset has to grow exponentially with the number of conditioning variables, such that conditioning on a lot of variables is often unreliable.

Things get worse when dealing with continuous variables. It now becomes impossible to perform $k$ separate tests because of continuity. Margaritis & Thrun (2001) propose an unconditional independence test for continuous variables, and build on it to propose a conditional independence test in Margaritis (2005) (the Recursive Median algorithm), both of which are very computationally expensive and cannot be used in a context where such procedures would have to be called at least a polynomial number of times.

Other techniques like conditional mutual information could be used with the property that a zero conditional mutual information is equivalent to conditional independence in the general case. This comes with an additional computational cost, however. Fleuret (2004) uses conditional mutual information for feature selection and discusses this additional cost for handling continuity. Bergsma (2004) discusses another technique using what he calls "partial copula."

Practically, testing for conditional independence between random variables is done using a test on

the vanishing *partial correlation*, which can be thought of as a kind of "conditional correlation." This concept is discussed in more details in Section 5.1.

### 4.6.4 Checking for *d*-Separation

The CBL algorithm tests for $d$-separation between two variables. This criterion is not so easy to deal with. There are two main approaches: a marking algorithm and an algorithm working with separation on a transformed undirected graph.

- **Bayes-Ball Algorithm**

  The original Bayes-Ball algorithm as described in Shachter (1998) takes a DAG and a query $p(X \mid Z)$ and determines three sets: the irrelevant nodes $\mathbf{N}_i(X \mid Z)$, the requisite probability nodes $\mathbf{N}_p(X \mid Z)$ and the requisite observation nodes $\mathbf{N}_e(X \mid Z)$. In order to answer the query, i.e, to find $p(X \mid Z)$, the irrelevant nodes are useless, the value of the requisite observation nodes must be known, and the conditional probability distribution of the requisite probability nodes must also be known.

  This algorithm allows to state that $X$ and $Y$ are $d$-separated given $Z$ by equivalently observing that $Y \in \mathbf{N}_i(X \mid Z)$.

  Starting from $X$, the Bayes-Ball algorithm marks the nodes according to whether they are visited from one of their children or one of their parents, and schedules accordingly its parents or children to be visited, depending on whether dependence can "flow" through the current node given the evidence $Z$. Its complexity is $\mathcal{O}(n)$. The constant factor can be reduced if we do not explicitly need to know $\mathbf{N}_p(X \mid Z)$ and $\mathbf{N}_e(X \mid Z)$.

- **Working on a Transformed Graph**

  Lauritzen *et al.* (1990) show that the query "are $X$ and $Y$ $d$-separated given $Z$ in the DAG $\mathcal{G}$" can be reduced to "are $X$ and $Y$ separated given $Z$ in $(\mathcal{G}_{\mathbf{An}(XYZ)^*})^m$." $\mathbf{An}(XYZ)^*$ is the set of all ancestors of $X$, $Y$ and $Z$, including $X$, $Y$ and $Z$ themselves, and $\mathcal{G}_{\mathbf{An}(XYZ)^*}$ is the subgraph of the DAG $\mathcal{G}$ containing only the nodes $\mathbf{An}(XYZ)^*$. Finally, $(\mathcal{G}_{\mathbf{An}(XYZ)^*})^m$ is the (undirected) moral graph, where edges are added between all parents of each node if they do not already exist.

  This problem can now be solved more easily: removing all edges from and to nodes in $Z$, we say that $X$ and $Y$ are separated if there is no remaining path between them. This can be checked by computing the reachability graph, which can also be done in linear time.

### 4.6.5 Finding the Minimal *d*-Separating Set

This is required by the CBL algorithm to help conduct conditional tests with the minimal number of conditioning variables. We present an exact algorithm and a greedy search version.

- **Acid-Campos Algorithm**

  Acid & de Campos (1996) have proposed an exact algorithm which works on $(\mathcal{G}_{\mathbf{An}(XY)^*})^m$ and returns one of the possible minimal sets of variables $d$-separating $X$ and $Y$. They have showed how it can be generalized to $d$-separating two sets of variables.

The algorithm draws its ideas from Ford and Fulkerson's maximal flow algorithm originally described in Ford & Fulkerson (1956). This algorithm works on edges and our goal is to return a set of nodes: Acid & de Campos (1996) show how to transform the graph. The proposed algorithm, however, directly works with the moral ancestor graph, marking nodes and edges according to a rather complicated set of rules and visiting order, which are not detailed here.

- **Greedy Search**

  Cheng *et al.* (1997a) propose a greedy search version, which is not guaranteed to find the minimal $d$-separating set (although all sets found on the ALARM network (Beinlich *et al.*, 1989) were reportedly actually minimal).

  It works by listing all open paths between the two nodes, and alternatively blocking the paths consisting of only one node and finding the node blocking the maximum number of longer paths, until no more open paths exist. When nodes have been added to the blocking set, a test is performed to check if this has opened new paths which must now be added to the list of open paths which need to be closed.

**Summary**

*We reviewed Bayesian and causal network structure learning algorithms. Search-and-score methods are generally used on the Bayesian side, whereas causal discovery usually prefers to construct the network under the conditional independence constraints. Strong assumptions like DAG-isomorphicity and Causal Sufficiency are needed to be able to derive relationships which we can say are causal. In some cases, latent variables can also be detected. Continuous variables are more difficult and more computationally expensive to deal with than discrete variables; missing values are also a problem.*

CHAPTER

5

# Alternative Approaches to Structure Learning

*In this chapter, we detail the concept of partial correlation, explain the intuition behind it and generalize it. We then present a new network construction algorithm based on partial correlation, which not only outputs a DAG pattern but also assigns weights to the edges, representing their respective robustness. The algorithm is tested on standard benchmarks.*

*We relate the feature selection problem to the causal network construction. Two approaches are discussed: using techniques from causality detection algorithms to do feature selection, and using techniques from feature selection to construct a causal graph.*

## 5.1 Partial Correlation

The concept of partial correlation is central to the new structure learning algorithm; it is presented here.

### 5.1.1 Definition & Intuition

The partial correlation was referred to in Subsection 4.6.3 as a kind of "conditional correlation" between two variables given a certain conditioning set. When conditioning on a set, we examine a certain property while holding the value of the conditioning variables constant: we want to suppress the effect of this variable.

This concept is termed *partial* correlation because what we want to measure is not the full correlation between two variables, but the remaining—partial—correlation after we remove the effect of the controlling variables. Suppose these controlling variables are continuous, and their effect has to be removed: what makes it possible for us to do that?

Let us jump directly to its definition and then interpret it.

**Definition 5.1.1 (Partial Correlation)** *The* partial correlation $\rho_{XY\cdot\mathbf{Z}}$ *between two variables* $X$ *and* $Y$ *given a set of controlling variables* $\mathbf{Z}$ *is the correlation between the residuals of the linear regression on* $X$ *using* $\mathbf{Z}$ *and the residuals of the linear regression on* $Y$ *using* $\mathbf{Z}$.

As a convention, we set $\rho_{XY\cdot\emptyset} = \rho_{XY}$. We can then define:

$$\rho_{XY\cdot\mathbf{Z}} = \text{corr}(R_X, R_Y) \tag{5.1}$$

with $R_X$, $R_Y$ being the respective residuals:

$$R_X = X - \langle \mathbf{w}_X^*, \mathbf{Z} \rangle \tag{5.2}$$
$$R_Y = Y - \langle \mathbf{w}_Y^*, \mathbf{Z} \rangle. \tag{5.3}$$

The coefficients $\mathbf{w}_\mathbf{X}^*$ and $\mathbf{w}_\mathbf{Y}^*$ are the solutions to the linear regression problems:

$$\mathbf{w}_X^* = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{p} (x_i - \langle \mathbf{w}, \mathbf{z}_i \rangle)^2 \right\} \tag{5.4}$$
$$\mathbf{w}_Y^* = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{p} (y_i - \langle \mathbf{w}, \mathbf{z}_i \rangle)^2 \right\} \tag{5.5}$$

where $\langle \mathbf{v}, \mathbf{w} \rangle$ is the scalar product between the vectors $\mathbf{v}$ and $\mathbf{w}$.

The key to understanding why this definition would be a good means to remove the effect of $\mathbf{Z}$ in the correlation lies in the interpretation of the regression. Regressing $X$ on $\mathbf{Z}$ can be seen as trying to best explain $X$ by $\mathbf{Z}$. The residuals $R_X$ are then the part of $X$ that $\mathbf{Z}$ cannot explain according to our model, i.e., the part of $X$ with the effect of $\mathbf{Z}$ removed. The same reasoning holds for $Y$ and $R_Y$.

Now, correlation between $R_X$ and $R_Y$ is then the correlation between the part of $X$ that cannot be explained by $\mathbf{Z}$ and the part of $Y$ that cannot be explained by $\mathbf{Z}$. It is indeed a good way to express the similarity between two while removing the effect of some controlling set.

Of course, it must be noted that we use *correlation* after *linear* regresssion, i.e., that we will get a similarity measure which will only detect linear dependencies after removing only the linear effect of the controlling variables. If we do not only assume that variables influence one another in a linear way but also want to allow other kinds of influences, this model is inappropriate. Some possible generalizations are discussed in Subsection 5.1.4.

Partial correlation has the interesting property that it can be less than, equal to or greater than the normal correlation (which is also called *zero-order* correlation).

### 5.1.2 Geometrical Interpretation

A geometrical interpretation of correlation and partial correlation helps getting the intuition behind it.

Choosing two variables $X$, $Y$ from our $n$-dimensional dataset $D$ of size $p$, we can regard their successive values over the samples as vectors $\mathbf{x}$ and $\mathbf{y}$. These vectors are $p$-dimensional and cannot be represented conveniently for $p > 3$; for the sake of simplicity, we use $p = 2$ in Figure 5.1 and $p = 3$ in Figure 5.2.

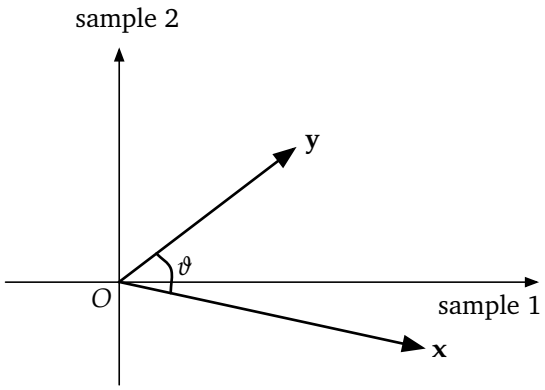



FIG. 5.1: The correlation between $X$ and $Y$ seen geometrically

If we want to express the similarity between these two vectors, we could for instance look at their norm, or at the angle between them. The cosine of the angle $\vartheta$ between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ can be written as

$$\cos\vartheta = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \tag{5.6}$$

$$= \frac{\sum_{i=1}^{p} x_i y_i}{\sqrt{\sum_{i=1}^{p} x_i^2} \sqrt{\sum_{i=1}^{p} y_i^2}}, \tag{5.7}$$

which is the same as the sample correlation, if we assume, without loss of generality, that the variables have a zero mean.

It can be shown that the residuals $R_X$ of regressing $X$ on $Z$, if also considered as a $p$-dimensional vector $\mathbf{r}_X$, have a zero scalar product with the vector $\mathbf{z}$ generated by $Z$: $\langle \mathbf{r}_X, \mathbf{z} \rangle = 0$. This means that the residuals vector lives on a hyperplane $S_{\mathbf{z}}$ which is perpendicular to $\mathbf{z}$, as shown in Figure 5.2.

The same also applies to the residuals $R_Y$ generating a vector $\mathbf{r}_Y$. The desired partial correlation is then the cosine of the angle $\varphi$ between the projections $\mathbf{r}_X$ and $\mathbf{r}_Y$ of $\mathbf{x}$ and $\mathbf{y}$, respectively, onto the hyperplane perpendicular to $\mathbf{z}$.

Through this interpretation, we see that we try to "remove the $Z$-dimension" from $X$ and $Y$ and then take the regular correlation between what remains once everything that had to do with $Z$ has been suppressed.

Further discussion and interpretation of the correlation and partial correlation can be found in Rummel (1976).

### 5.1.3 Efficient Computation

If we choose partial correlation as our conditional similarity measure, we must ensure that it can be computed in an efficient way. One possibility is the hard way, solving the linear regression problem, getting the residuals and calculating the correlation. This is not very efficient, but fortunately, it turns out that $k$th-order partial correlation can be computed from $(k-1)$th-order partial

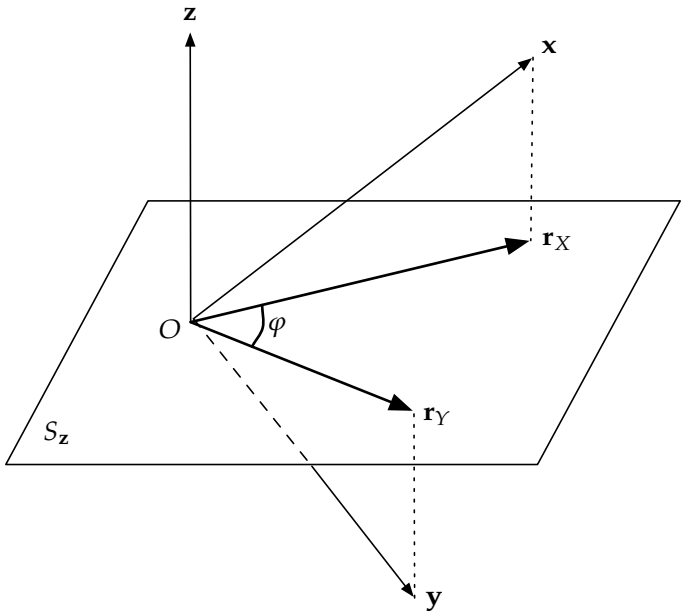



FIG. 5.2: The partial correlation between $X$ and $Y$ given $Z$ seen geometrically

correlation using the recursive formula:

$$\rho_{XY\cdot\mathbf{Z}\cup\{W\}} = \frac{\rho_{XY\cdot\mathbf{Z}} - \rho_{XW\cdot\mathbf{Z}}\rho_{YW\cdot\mathbf{Z}}}{\sqrt{1-\rho^2_{XW\cdot\mathbf{Z}}}\sqrt{1-\rho^2_{YW\cdot\mathbf{Z}}}}. \tag{5.8}$$

The proof for the special case $\mathbf{Z} = \emptyset$ is given in Appendix B.

Solving a partial correlation problem of size $k$ is then equivalent to solving three problems of size $k-1$, as far as complexity is concerned. A naïve implementation will yield an exponential complexity; see the example call graph in Figure 5.3.

Two observations allow us to reduce this complexity:

- Partial correlation is symmetrical with respect to the non-controlling variables $X$ and $Y$, like zero-order correlation: $\rho_{XY\cdot\mathbf{Z}} = \rho_{YX\cdot\mathbf{Z}}$;
- By definition, the order of the variables in the controlling set does not matter, such that, for instance, $\rho_{XY\cdot ZW} = \rho_{YX\cdot WZ}$.

Caching the intermediate results from the recursive calls and rearranging the order of variables in the controlling set (which we implement as ordered list or array in practice) let us reduce the complexity to $\mathcal{O}(k^3)$. The modified call graph integrating these considerations is shown in Figure 5.4.

Similarly, naïvely implementing the computation of partial correlation for all variable pairs yields a $\mathcal{O}(n^5)$ complexity, but caching also improves it. Numerical simulation suggests a complexity not greater than $\mathcal{O}(n^4)$, although this is not proven yet.

Although the complexity, whether it is $\mathcal{O}(n^5)$ or $\mathcal{O}(n^4)$, is important and renders this approach

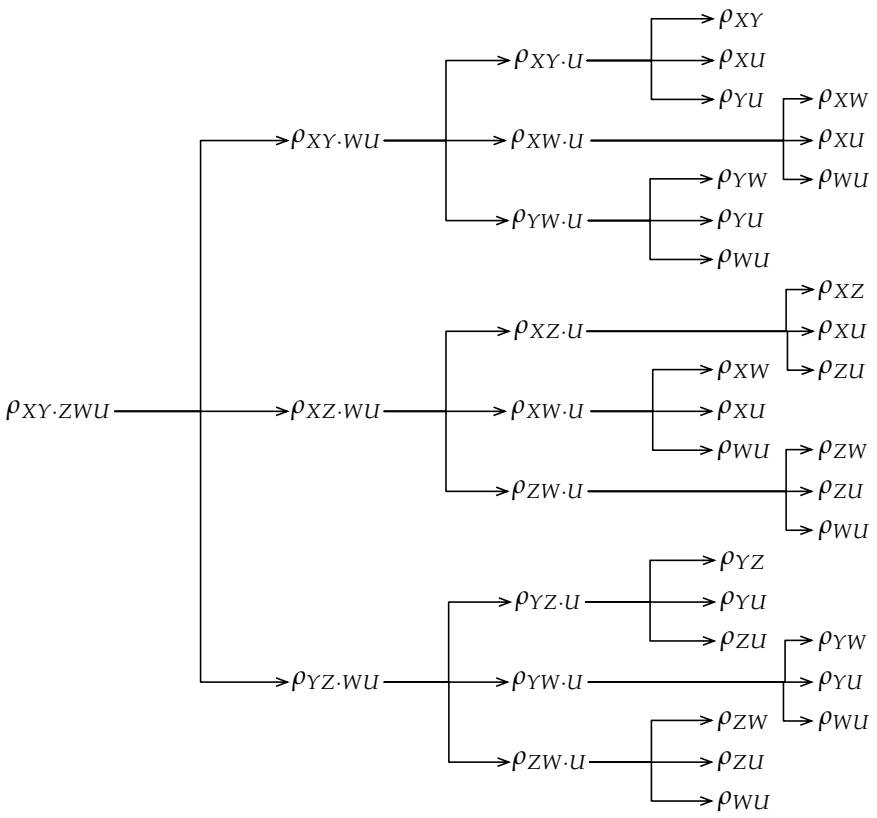



FIG. 5.3: Naïve call graph for a partial correlation problem of size $k = 3$: 40 calls

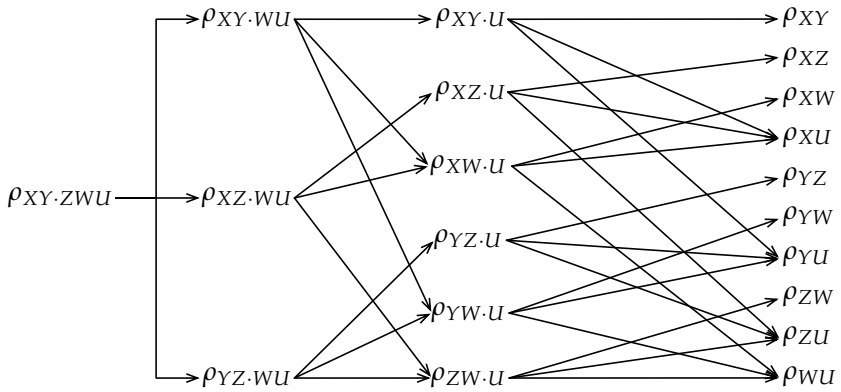



FIG. 5.4: Call graph implementing caching for a partial correlation problem of size $k = 3$: 20 calls

hardly scalable, this is also the typical complexity found for instance in the self-orienting CBL algorithm. Both the PC and the IC* algorithms have a higher complexity.

What is also to be considered is the complexity in terms of the sample size $p$. For now, keeping $n$ constant, all considered approaches have a linear complexity $\mathcal{O}(p)$.

Another approach (Schäfer & Strimmer, 2005) allows to compute in $\mathcal{O}(n^3)$ all partial correlations between any two variables $X$ and $Y$ of a set $\mathbf{V}$ of cardinality $n$ given all others, i.e., $\mathbf{V} \setminus \{X, Y\}$, provided the correlation matrix $\mathbf{\Omega} = (\omega_{ij})$ where $\omega_{ij} = \text{corr}(X_i, X_j)$ is invertible. If we define $\mathbf{P} = \mathbf{\Omega}^{-1}$, we have:

$$\rho_{X_i X_j \cdot \mathbf{V} \setminus \{X_i, X_j\}} = -\frac{p_{ij}}{\sqrt{p_{ii} p_{jj}}}. \tag{5.9}$$

When computing several partial correlations, it may turn out that the union of the controlling and non-controlling variables is the same for several calls, so that here too, we can use caching and improve the computation time (although not the overall complexity). In practice and in context of the algorithm described below, this method proves most efficient.

### 5.1.4 Generalization

As stated previously, the main restriction of the partial correlation as defined here is that we use correlation, a linear similarity measure, after linear regresssion. To try and relax these linearity assumptions, we can try to replace the regression and similarity steps by nonlinear equivalents. We have indeed no indication tending to show that causal relationships are always linear.

The linear regression is traditionally solved using the least mean squares solution, which requires inverting the covariance matrix. To help solve badly-conditioned linear regression problems, i.e., problems where inverting the covariance matrix leads to numerical difficulties, we can use ridge regression (Hoerl & Kennard, 1970). Both linear regression and ridge regression can benefit from the "kernel trick" (Aizerman *et al.*, 1964) in order to gain nonlinear capabilities.

Let $\mathbf{x}_i$, $1 \leq i \leq p$, be the $i$th $n$-dimensional sample of our regressor set, and let $k(\mathbf{x}, \mathbf{x}')$ be a kernel function subject to Mercer's condition (Mercer, 1909), corresponding to a scalar product $\langle \Phi(x), \Phi(x') \rangle$ after a nonlinear mapping $\Phi$ into some higher-dimensional space. If we define an $p \times p$ matrix $\mathbf{K} = (k_{ij})$ with $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, it can be shown (see Appendix B) that the optimal weight vectors $\mathbf{w}_L^*$ and $\mathbf{w}_R^*$ for linear and ridge regression, respectively, are

$$\mathbf{w}_L^* = \mathbf{X}^T \mathbf{K}^{-1} \mathbf{y} \tag{5.10}$$

$$\mathbf{w}_R^* = \mathbf{X}^T (\mathbf{K}^2 + \lambda \mathbf{K})^{-1} \mathbf{K} \mathbf{y} \tag{5.11}$$

with $\lambda$ being the ridge value, $\mathbf{X}$ a $p \times n$ regressor matrix and $\mathbf{y}$ the $p$-dimensional vector representing the variable to regress.

In both cases, nonlinearity comes with a high price: this matrix inversion leads to a complexity of $\mathcal{O}(p^3)$, $p$ being still the number of *samples* and not variables. Finding a closed-form formula for the partial correlation given a certain kernel which would bypass this matrix inversion would be very convenient; as of now, computing the conditional correlation for all variable pairs with this technique has a complexity of $\mathcal{O}(p^3 n^3)$ in the worst and average cases.

Other forms of regression, like Support Vector Regression (Smola & Schölkopf, 1998) or $k$-Nearest-

Neighbors regression (Devroye *et al.*, 1994), could be used. These extensions were not part of this study but can be explored in future work.

The similarity measure to be computed on the residuals can also be changed to another nonlinear method like mutual information. However, the method will have to do with continuous data and the considerations described in Subsection 4.6.1 apply.

## 5.2 An Algorithm Based on Partial Correlation

Having introduced partial correlation, we now motivate and describe a new algorithm based on it. We make the standard assumptions of DAG-isomorphism and Causal Sufficiency.

### 5.2.1 Partial Correlation & Conditional Independence

The semantics of our graphs, and in particular the P-map property, link the $d$-separation criterion to conditional independence. The question arises to know whether and when partial correlation can be used as a test for conditional independence. Baba *et al.* (2004) discuss when the two concepts can be equated, and conclude that when the joint probability distribution is known to be multivariate Gaussian, then a zero partial correlation is equivalent to statistical independence.

Why use partial correlation in the first place? For one, it is easy to condition on a lot a variables—respectively, to include many variables in the controlling set—without reducing the power of the test as with multiple $\chi^2$ tests. Moreover, it works with continuous variables in an efficient way.

Note that the generalizations of partial correlation presented in Subsection 5.1.4, while they may allow for nonlinear "explanations" by the controlling variables, may or may not preserve the equivalence to conditional independence under the Gaussian assumption. This remains to be investigated.

Another benefit that we get using partial correlation is that we do not only get a boolean answer with a p-value such as what the $\chi^2$ test returns, but also a continuous range from $-1$ to 1 describing the degree of association of the two variables. We make use of this feature in the algorithm we develop below; this can also be used to give a weight to the edges and arcs in our final graph to distinguish those which are strong from those which could easily vanish had we examined a slightly different dataset. The benefit of using the value of the partial correlation as weight rather than a p-value is that they coincide with the path coefficients in a linear SEM. Assigning weights to links indicates the *robustness* of these links. This concept is to be related to the work by Friedman *et al.* (1999) on robust structure learning.

In order to test for a vanishing partial correlation, we use Fisher's $z$-transform on the sample partial correlation:

$$z(\hat{\rho}_{XY\cdot\mathbf{Z}}) = \frac{1}{2}\ln\left(\frac{1+\hat{\rho}_{XY\cdot\mathbf{Z}}}{1-\hat{\rho}_{XY\cdot\mathbf{Z}}}\right). \tag{5.12}$$

The null hypothesis is $H_0 : \hat{\rho}_{XY\cdot\mathbf{Z}} = 0$, to be tested against $H_A : \hat{\rho}_{XY\cdot\mathbf{Z}} \neq 0$. We reject $H_0$ with significance level $\alpha$ if

$$\sqrt{n - |\mathbf{Z}| - 3}\cdot z(\hat{\rho}_{XY\cdot\mathbf{Z}}) > \Phi^{-1}(1-\alpha/2), \tag{5.13}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a Gaussian distribution with zero mean and unit standard deviation.

### 5.2.2 Principles of the Algorithm

Our algorithm will work in two phases, like PC or IC*: it will first build the undirected structure of the graph, and then orient some edges with V-structure detection and constraint propagation. Let us remember that in a P-map, a link between two nodes $X$ and $Y$ exists if and only if $X$ and $Y$ are conditionally dependent, whatever the conditioning set is. With $n$ variables, testing naïvely this condition leads to conducting $2^{n-2}$ conditional independence tests for each undirected variable pair, i.e., $\frac{n(n-1)}{2}2^{n-2}$ tests in total. Our goal is to find a smart way to conduct a smaller number of tests, or ideally, only one per variable pair.

A first idea would be to condition on all other variables (i.e., do an analysis between two variables, *ceteris paribus*) and to keep only the links corresponding to the non-vanishing partial correlations. We would thus remove the effect of all variables other than the two in the pair we are considering. This is what is done in the framework of graphical Gaussian models, briefly discussed in Schäfer & Strimmer (2005). Although this may sound reasonable, we must pay attention to the fact that these models use undirected graphs with the separation criterion, and that our DAGs with $d$-separation do not allow us to use such an approach.

This can be illustrated by this example: suppose the true structure of some problem to be $X \rightarrow Z \leftarrow Y$. If, for each variable pair, we conduct a conditional independence test conditioning on the remaining variables, we will get the following set of dependencies:

$$(X \not\perp\!\!\!\perp Y \mid Z) \qquad (X \not\perp\!\!\!\perp Z \mid Y) \qquad (Y \not\perp\!\!\!\perp Z \mid X)\,, \tag{5.14}$$

which does not allow us to draw anything but the fully connected graph, thus missing the fact that $(X \perp\!\!\!\perp Y)$ holds. Note, however, that the fully connected graph actually represents the minimal undirected I-map of the data (and is the moral graph of the original structure).

This, again, has to do with V-structures. Conditioning on colliders between two variables actually opens a dependence path, whereas it blocks it for divergent nodes or nodes in unidirected chains. If we want to condition on as many variables as possible, we have to exclude the colliders. But we have the problem that colliders between two variables $X$ and $Y$ can be identified with traditional conditional independence tests only if we already know of a set which makes $X$ and $Y$ independent. In the previous example, this was the empty set, but consider the example structure in Figure 5.5.

To formally identify $Z_1$ and $Z_2$ as colliders for the pair $(X, Y)$, we need to know that $\{U, W_1, W_2\}$ is a $d$-separating set, i.e., that $(X \perp\!\!\!\perp Y \mid \{U, W_1, W_2\})$, and then see that the following holds:

$$(X \not\perp\!\!\!\perp Y \mid \{U, W_1, W_2, Z_1\}) \qquad (X \not\perp\!\!\!\perp Y \mid \{U, W_1, W_2, Z_2\})\,. \tag{5.15}$$

With traditional tests, we have no *a priori* way to distinguish a $W_i$ or a $U$ from a $Z_i$. In this particular situation, we would have to conduct again an exponential number of tests to identify the colliders.

But with partial correlation, we get levels of correlations. Conditioning on certain nodes can make the correlation less or greater than the zero-order correlation. Looking at the absolute value of correlation as a measure of dependence, we could intuitively expect conditioning on a collider

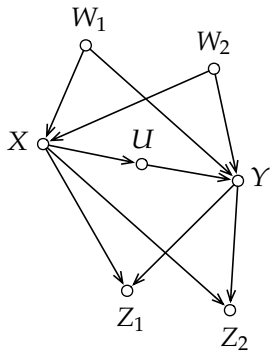



FIG. 5.5: An example structure to illustrate collider identification

to strengthen the correlation and conditioning or a divergent node or chained node to weaken it, independently of which variables we condition on along the other paths linking $X$ to $Y$. For instance, for this example, we expect:

$$\begin{aligned} |\rho_{XY\cdot W_1}| &\leq |\rho_{XY}| & |\rho_{XY\cdot Z_1}| &\geq |\rho_{XY}| \\ |\rho_{XY\cdot W_2}| &\leq |\rho_{XY}| & |\rho_{XY\cdot Z_2}| &\geq |\rho_{XY}| \\ |\rho_{XY\cdot U}| &\leq |\rho_{XY}|. \end{aligned} \tag{5.16}$$

Further reasoning in this direction, we can make the following hypothesis.

**Hypothesis 5.2.1** *For a DAG-isomorphic joint probability distribution $p(\mathbf{V})$, the following holds for distinct $X$, $Y$, $W \in \mathbf{V}$ and $\mathbf{Z} \subset \mathbf{V}$:*

$$|\rho_{XY\cdot W}| > |\rho_{XY}| \implies |\rho_{XY\cdot \mathbf{Z}\cup\{W\}}| \geq |\rho_{XY\cdot \mathbf{Z}}| \tag{5.17}$$

$$|\rho_{XY\cdot W}| < |\rho_{XY}| \implies |\rho_{XY\cdot \mathbf{Z}\cup\{W\}}| \leq |\rho_{XY\cdot \mathbf{Z}}|. \tag{5.18}$$

The conditions, if any, under which this hypothesis holds, remain to be defined and will be looked into in future work. What this hypothesis embodies is the intuition that a collider increases the correlation when conditioned on, whatever other variables are already in the controlling set. In the worst case, it will have no effect but will not suddenly start behaving as if it were a common cause or chained node. We also want the converse to hold: a common cause or a chained node will not start acting like a collider depending on which other nodes we condition on.

If we accept Hypothesis 5.2.1, we can state the following corollary.

**Corollary 5.2.2** *The conditioning set $\mathbf{S}^{max}$ for two variables $X$ and $Y$ which yields the highest absolute partial correlation is the set of all nodes increasing the absolute first-order partial correlation with respect to the absolute zero-order correlation:*

$$\begin{aligned} \mathbf{S}^{max}_{XY} &= \arg\max_{\mathbf{S}\subset\mathbf{V}} |\rho_{XY\cdot\mathbf{S}}| \\ &= \{Z \mid |\rho_{XY\cdot Z}| \geq |\rho_{XY}|\}. \end{aligned} \tag{5.19}$$

*Similarly, the conditioning set $\mathbf{S}^{min}$ for two variables $X$ and $Y$ which yields the lowest absolute partial correlation is the set of all nodes reducing the absolute first-order partial correlation with respect to*

*the absolute zero-order correlation:*

$$\begin{aligned} \mathbf{S}_{XY}^{min} &= \arg\min_{\mathbf{S}\subset\mathbf{V}} |\rho_{XY\cdot\mathbf{S}}| \\ &= \{Z \mid |\rho_{XY\cdot Z}| \leq |\rho_{XY}|\}. \end{aligned} \tag{5.20}$$

### 5.2.3 The Algorithm

We have now found a way to replace the conditioning on all variables, which is correct for undirected graphs, with a conditioning on a more relevant set of variables, which takes into account the particularities of the $d$-separation criterion, and we can use this to draft our structure construction algorithm. Pseudocode can be found in Algorithm 1 and is repeated in Appendix A for consistency.

**Algorithm 1** PartCorr construction algorithm

**procedure** PARTCORRCONSTR
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ empty graph with $d$ nodes
normalize $D$ so that variables have a zero mean
**for each** u **do** ndirected pair $(X, Y)$: $coll_{XY} \leftarrow \emptyset$

/* Undirected structure construction */
**for each** undirected pair $(X, Y)$ **do**
$coll_{XY} \leftarrow$ the set of all nodes $Z \neq X, Y$ such that $|\rho_{XY\cdot Z}| > |\rho_{XY}|$
$\mathbf{S}_{XY} \leftarrow \mathbf{V} \setminus coll_{XY} \setminus \{X, Y\}$
**if** $\rho_{XY\cdot\mathbf{S}_{XY}}$ does not vanish **then**
add an undirected edge $X - Y$ to $\mathcal{G}$
**end if**
**end for**

/* V-structure detection */
**for each** $X, Y, Z$ such that $X - Z - Y$ **do**
**if** $Z \notin \mathbf{S}_{XY}$ **then** orient as $X \rightarrow Z \leftarrow Y$
**end for**

/* Constraint propagation */
**while** $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
**for each** $X, Y, Z$ such that $X \rightarrow Y - Z$ **do**
orient as $X \rightarrow Y \rightarrow Z$ /* no new V-structure */
**end for**
**for each** $X, Y$ such that $X - Y$ and $\exists$ directed path from $X$ to $Y$ **do**
orient as $X \rightarrow Y$ /* preserve acyclicity */
**end for**
**for each** $X, Y$ s.t. $X - Y$ and $\exists$ nonadjacent $Z, W$ s.t. $X - Z \rightarrow Y$ and $X - W \rightarrow Y$ **do**
orient as $X \rightarrow Y$ /* three-fork V with married parents */
**end for**
**end while**
**end procedure**

**Theorem 5.2.3** *Assuming a multivariate Gaussian joint probability distribution of the variables, the phase "Undirected structure construction" of the PartCorr algorithm constructs the correct undirected structure.*

**Proof** With a multivariate Gaussian distribution, a partial correlation $\rho_{XY\cdot\mathbf{Z}}$ vanishes if and only if $X$ is independent of $Y$ given $\mathbf{Z}$. Moreover, we know that an edge must be added between $X$ and $Y$ if and only if $X$ and $Y$ are dependent conditioned on any possible subset of $\mathbf{V} \setminus \{X, Y\}$. We must then prove the following:

$$\left(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\right) \iff X \text{ and } Y \text{ are connected after Phase 1,} \tag{5.21}$$

which is equivalent to

$$\left(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \perp\!\!\!\perp Y \mid \mathbf{S})\right) \iff X \text{ and } Y \text{ are } \textit{not} \text{ connected after Phase 1.} \tag{5.22}$$

If we can find a single set for which the partial correlation between $X$ and $Y$ vanishes, we know that we must not add an edge between $X$ and $Y$. So, if we look at the conditioning set having the lowest absolute partial correlation and we find that the partial correlation is not negligible, it means that there is no set for which it vanishes and thus that an edge must exist between $X$ and $Y$. According to Corollary 5.2.2, this set is the set $\mathbf{S}_{XY}^{min}$ of all nodes $Z$ which satisfy $|\rho_{XY\cdot Z}| \leq |\rho_{XY}|$. It corresponds to the set $\mathbf{S}_{XY}$ as computed by Algorithm 1, namely $\mathbf{V} \setminus coll_{XY} \setminus \{X, Y\}$. The edge is then added if and only if this partial correlation does not vanish. This proves Requirement 5.21 and thus completes the proof. □

Intuitively, we want to build a set which makes the partial correlation as small as possible. If, despite our efforts, the partial correlation still does not vanish, then a true link must exist between $X$ and $Y$.

### 5.2.4 Benchmarks

We tested this algorithm with the same set of tests used throughout Chapter 4. The constructed graphs are shown in Figure 5.6; run times were 1.42 s, 0.59 s, 0.56 s and 1.60 s, respectively. The varying run times come from a lower or higher number of cache hits during partial correlation computation.

Although the principles of this algorithm are based on conditional independence tests, we see that it does not suffer from small datasets as much as IC/PC or CBL. It does need more samples to find the correct graph (up to the missing weak link), but also correctly directs some arcs with $p = 50$ already, where PC/IC made a mistake.

It fails at the XOR test because each pair of nodes determines that the remaining one is a collider and then compute three zero-order correlations to test for the existence of direct links—which again are zero. It is thus also vulnerable to violations of the DAG-isomorphicity assumption.

Looking promising, the algorithm was tested with a bigger problem, the ALARM network (Beinlich *et al.*, 1989). This network was originally designed to help interpret monitoring data to alert anesthesiologists to various situations in the operating room. It has 37 variables and 46 edges, 4 of which are undirected in the equivalence class (they are drawn in bold in Figure 5.7).

The graph was used as a structure for a linear SEM. The parentless variables were sampled as Gaussians with zero mean and unit standard deviation; the other variables were defined as a

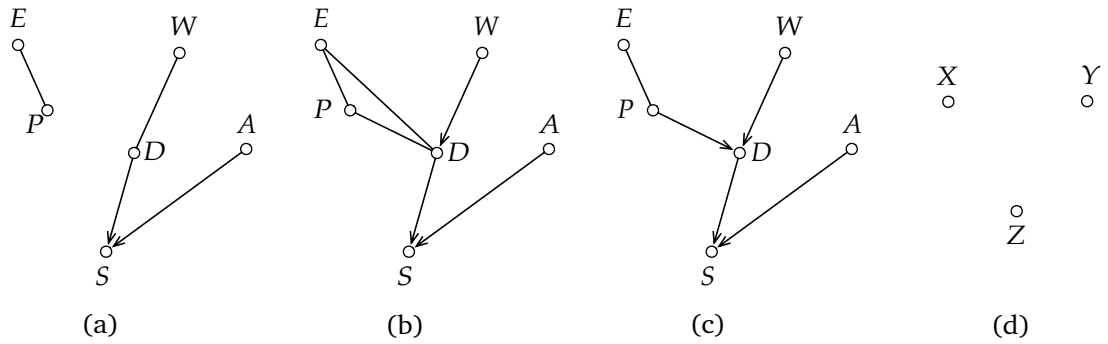



FIG. 5.6: Results for PartCorr: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$; (d) XOR, $p = 1000$

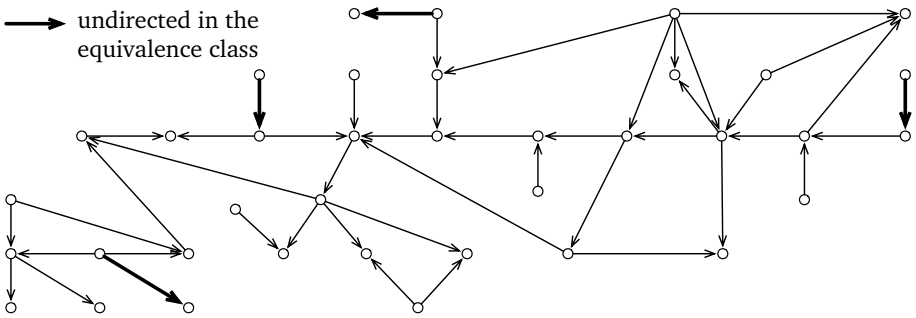



FIG. 5.7: The original ALARM network

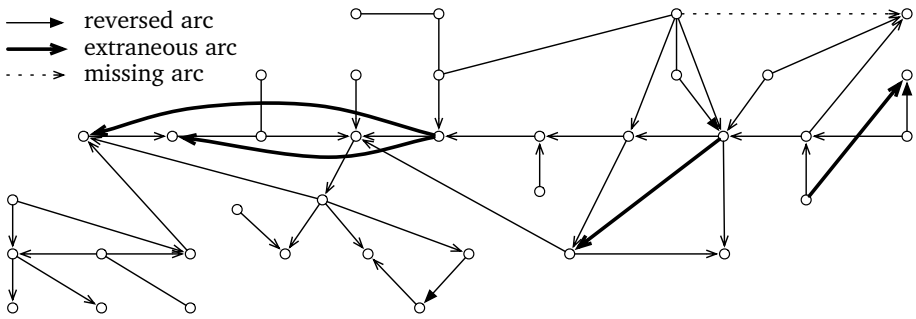



FIG. 5.8: The recovered ALARM network with $p = 1000$

linear combination of their parents with coefficient randomly distributed between 0.1 and 0.9, similarly to what was done in Scheines *et al.* (1995). The disturbance terms were also normally distributed. The statistical significance of the statistical tests was set to $\alpha = 0.02$. The network was sampled $p = 1000$ times.

The PartCorr algorithm terminated after 47.12 s. The recovered graph is shown in Figure 5.8. There are 4 extraneous arcs (bold), 1 missing arc (dashed) and 4 wrongly directed arcs (with filled arrows).

Algorithms like IC/PC have shown slightly better scores, making typically 2 or 3 mistakes in the undirected structure where we make 5 and none on the link direction (Scheines *et al.*, 1995). CBL misses between 0 and 2 links with $p = 10000$ (Cheng *et al.*, 1997a). However, this benchmark makes us confident enough about our algorithm to use it as main tool for analyzing the real-world data in Chapter 6. It also tends to back Hypothesis 5.2.1 as no big deviation from the original graph structure can be systematically observed.

A very simple modification of this algorithm lets it handle problems with missing values more effectively than other approaches deleting the incomplete data rows. Indeed, every partial correlation can be ultimately calculated with zero-order correlations according to Equation 5.8 or Equation 5.9. When calculating a zero-order correlation between $X$ and $Y$, we only need to delete rows where $X$ or $Y$ is missing, regardless of whether or not other variables are missing in the data row, thus reducing the probability of a data row being removed to $p_m = 1 - (1 - m)^2$ rather than $1 - (1 - m)^n > p_m$, assuming a uniform probability $m$ of a value missing. Note that this approach is only valid for the MCAR missingness type and is biased for MAR or NMAR.

We tested the modified algorithm on the same set of problems as Structural EM. The run times were very similar to the same problem without missing values, as only the initial short phase of correlation computation changed. The results are shown in Figure 5.9 and the errors summarized in Table 5.1.

| Frequency of missing values | Toy Retail $p = 50$ | Toy Retail $p = 200$ | Toy Retail $p = 1000$ |
|---|---|---|---|
| $m = 10\%$ | 2, 0 (0) | 1, 0 (0) | 1, 0 (0) |
| $m = 25\%$ | 2, 2 (0) | 1, 1 (1) | 1, 1 (0) |
| $m = 50\%$ | 4, 3 (0) | 1, 2 (1) | 0, 3 (1) |

Table 5.1: Errors of PartCorr on the Toy Retail dataset with missing values

Although the behavior of the algorithm remains a bit erratic with $m = 50\%$, we can still recognize part of the structure. Whereas Structural EM tended to detect no links at all, PartCorr tends to detect too many of them. With $m = 10\%$, PartCorr gives the same output as when it was run with the complete dataset, except for (b), where it even surprisingly (and probably, by chance) corrects a wrongly added edge between $E$ and $D$ in Figure 5.6.

## 5.3 Relations to Feature Selection

The problem of causal network construction and the problem of variable or feature selection are not unrelated. We briefly restate the latter, and discuss how techniques from one can be used for the other, underlining potential caveats.

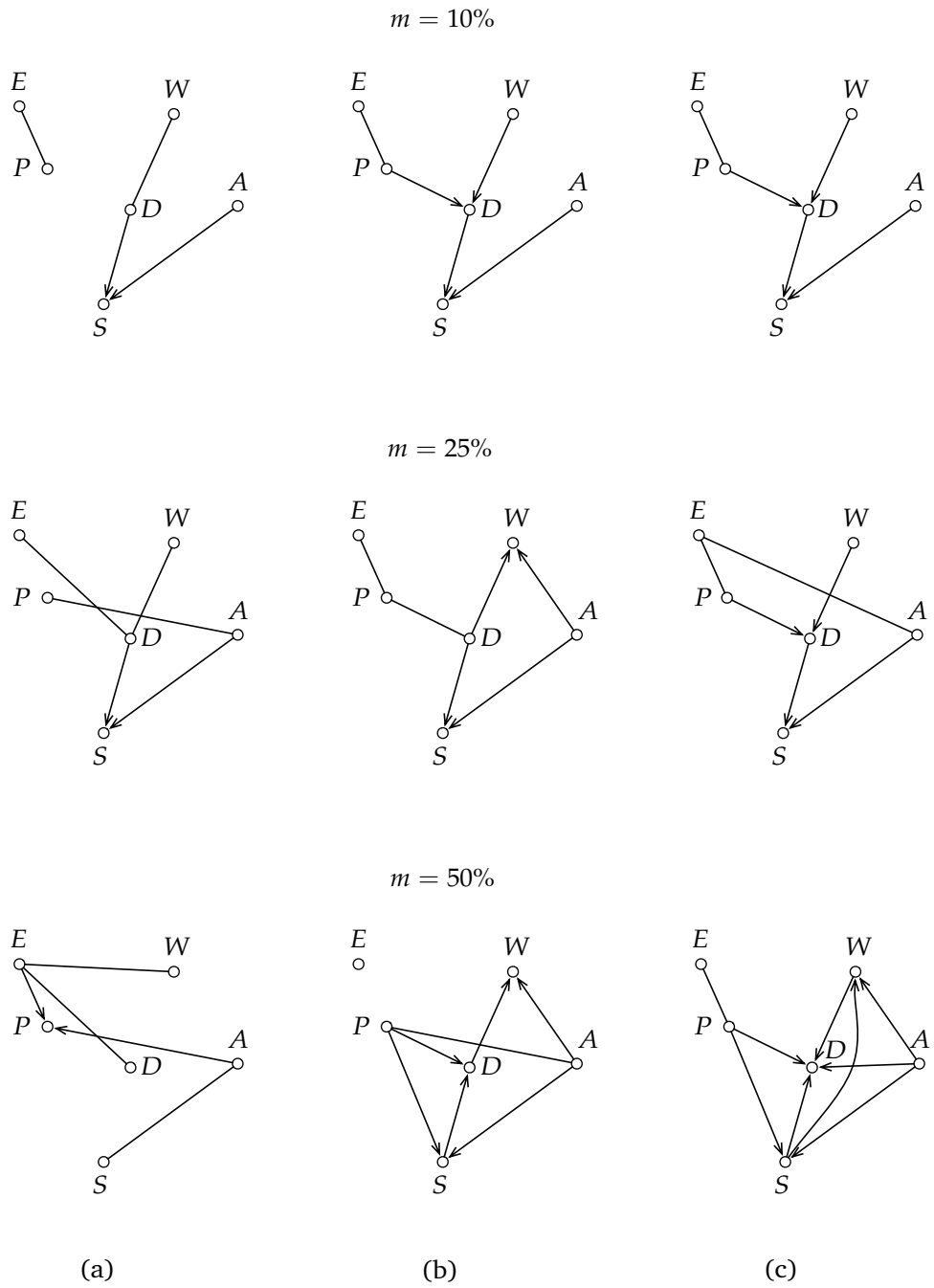



FIG. 5.9: Graphs for PartCorr with missing data: Toy Retail, (a) $p = 50$, (b) $p = 200$, (c) $p = 1000$

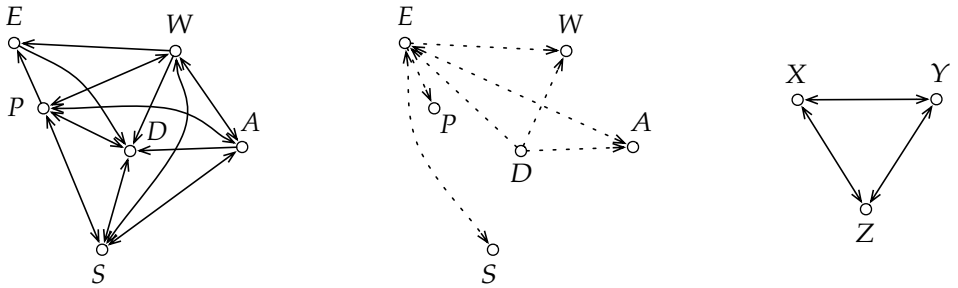
The feature selection problem can be stated as follows: given a dataset $D = \{(\mathbf{x}^i, y^i) \mid 1 \leq i \leq p\}$ with $\mathbf{x}^i = (x_1^i, x_2^i, \cdots, x_n^i)^T$, find the optimal subset of variables $\mathbf{S}_Y \subset \mathbf{X}$ (possibly of a pre-determined cardinality $m$) which allows to predict $Y$ optimally. Feature selection is often a mandatory step to reduce the problem dimension in datasets where the number of variables $n$ can reach several thousands. Good tutorials on feature selection include Guyon & Elisseeff (2003) and Kohavi & John (1997).

### 5.3.1 Causation Detection with Feature Selection

In feature selection and machine learning, there is a long tradition of using nonlinear techniques to detect more functional relationships or dependencies in the data. We can try to leverage that in our structure construction problem, where one of the issues is to break free from the linearity constraint without making the computational time explode.

One of the problems, however, is to automatically detect $m$, the cardinality of the set of predictors $\mathbf{S}_Y$ for a variable $Y$. Obviously, it cannot be set *a priori*, unlike other specific applications of feature selection. However, what we can do in a preliminary attempt to find a solution is to train some regression algorithm with optimal predictor sets of size $m$ with $1 \leq m \leq n-1$. A feature selection algorithm called Recursive Feature Elimination (RFE—see Guyon *et al.* (2002) for details) can be used, and the best generalization error across the $n-1$ trainings would determine the optimal number of predictors $m^*$.

We did a few preliminary tests on the Toy Retail and the XOR datasets using the Spider Matlab library (Weston *et al.*, 2003). Our main regression algorithm was a kernelized least mean squares regressor with a polynomial kernel of degree 2. RFE was used on it $n-1$ times for each of the $n$ variables; a 5-fold cross-validation was used to measure the generalization error. For Toy Retail, we reduced the number of samples to $p = 200$ only to keep the run time acceptable (6 min 41 s—with a kernelized regression engine, recall that we have a cubic complexity in terms of the number of samples); for XOR, we kept $p = 1000$ because of the small number of samples (the run time was 29 min 3 s). The results are shown in Figure 5.10: (a) and (c) shows an arc $X \to Y$ whenever variable $X$ was selected in the set of predictors for variable $Y$; (b) shows the difference between (a) and the complete bidirected graph for the same problem.




(a) (b) (c)

FIG. 5.10: Results using feature selection: (a) & (b) Toy Retail, $p = 200$; (c) XOR, $p = 1000$

Links will in principle be created not only with parents and children of $Y$ but also with its children's parents, because this forms the set of variables that makes $Y$ independent of other variables. But even taking this into account, we get a graph that is hardly sparse enough to be of any use.

Moreover, almost all arcs are bidirected, which tells us little about causation. One of the main problems is the phenomenon that redundant variables are likely to be all selected because they can significantly increase the predictive power (Guyon & Elisseeff, 2003). We hope to make progress on this issue in the future, shedding more light on how to use similar techniques in a sound way and reducing the very high computational time required by this naïve implementation.

### 5.3.2 Feature Selection with Causation Detection

Seeing things the other way, we might want to use techniques from causal network construction to do a better job at feature selection.

The solution to the feature selection problem for predicting a certain variable $Y$ is a set of variables $\mathbf{S}_Y$ allowing to obtain optimal predictions. Compare it to the definition of the Markov blanket of a variable in a causal network: its parents, children, and children's parents. With the $d$-separation criterion, this is the set of variables which makes $Y$ conditionally independent from all others. That is, knowing the value of more variables than those in the Markov blanket does not change anything in the probability distribution of our variable of interest: it does not help predicting it better.

Stated like this, the Markov blanket is the optimal solution to the feature selection problem. But this property of Markov blanket in a causal graph only holds under a set of assumptions, the most important of which is probably the DAG-isomorphicity assumption. If we are reasoning on a P-map, then the Markov blanket is the optimal solution. If we are reasoning on an I-map for some reason—lack of DAG-isomorphicity for instance—then we will, in the general case, include more variables than actually needed. Using a D-map would omit potentially relevant variables. The problem is that, to the best of our knowledge, no algorithm exists which would provably find at least a non-trivial I-map, possibly the minimal I-map, for a problem like the XOR scenario.

This is a big issue in feature selection, as variables can have more predicting power when taken together. The interested reader is invited to look at Guyon & Elisseeff (2003) for a discussion of this issue, which leads to the discussion of forward vs. backward selection. Looking at the Markov blanket of a causal graph built using only pairwise (conditional) similarity measures is comparable to forward feature selection.

Some work has been done for DAG-isomorphic problems, however. Aliferis *et al.* (2003) and Tsamardinos *et al.* (2003) discuss an algorithm which induces the local Markov blanket of a given variable. It works also with a very large number of variables $n$ as it is not necessary to construct the whole network. Building on this, Tsamardinos *et al.* (2006) introduce a new algorithm for learning the complete structure, using a mixture of search-and-score and constraint-based techniques. The technical report also includes a very detailed systematic comparison of several structure learning algorithms.

**Summary**

*Under certain assumptions, partial correlation can be used as a measure of conditional independence between variables. It can be computed efficiently and is especially useful with continuous variables. It is used successfully in the PartCorr algorithm, which can thus assign meaningful weights to the links it detects. Preliminary benchmarks show that the algorithm's idea is promising. Other approaches to causation detection could use techniques from feature selection methods, but are likely to make the computational time and complexity explode; on the other hand, the optimal solution to feature selection for a variable to be optimally predicted can be regarded as its Markov blanket and can therefore be found with causal structure learning techniques.*

CHAPTER

6

# Experimental Results

*We analyze real-world retail data with causal algorithms. Data from five different stores and for several hundred products are available. We use PartCorr to build causal graphs with weighted links, and compare its output with the results of a modified version of the PC algorithm. The results of several experimental scenarios are discussed and commented.*

## 6.1 Data Structure Description

The data we have at our disposal describes the sales of a European do-it-yourself chain of retail stores. We have data from $m = 5$ different retail stores, each dataset describing the sales of $k = 510$ products with a unique ID over a time period of $p = 556$ days. The exact nature of the products is unknown.

For each location, an independent list of promotions can be accessed. Promotions also have a unique ID. They can be in the form of flyers distributed to households, in which case they are usually associated with several products, or can be a rebate of some kind for only one product or a bundle of products, but we do not have access to this distinction. Begin date and end date are available for each promotion.

Additionally, detailed information about the weather in each location is available in the form of $q = 5$ variables: the minimum temperature, the maximum temperature, the rain or snow quantity in $\mathrm{mm}/\mathrm{cm}^2$, the number of sunny hours in the day, and the height of the snow, if any, at a certain fixed point in the day.

Practically, the raw data is split across several files: tab-delimited ASCII text files for the transactions (actually just a text dump of a database table), and Excel or comma-separated values format for the promotions and weather information. Python scripts and Matlab commands and scripts reformat it and arrange it into the structure described below.

**Notation** Formally, we will work with four matrices for each location $\ell$.

- $\mathbf{S}^\ell$: $p \times k$ matrix ($\mathbf{S}$ for *Sales*) where $s_{ij}^\ell$ is the sum of transactions on day $i$ for product $j$;
- $\mathbf{M}^\ell$: $p \times k$ binary matrix ($\mathbf{M}$ for *Marketing*) where $m_{ij}^\ell$ is 1 if some promotion is running on day $i$ for product $j$ and 0 otherwise;
- $\mathbf{W}^\ell$: $p \times q$ matrix where $w_{ij}^\ell$ is the value of the $j$th weather feature on day $i$;
- $\mathbf{T}^\ell$: $p \times r$ matrix $r = 4$ ($\mathbf{T}$ for *Time*) where $t_{ij}^\ell$ is the value of the $j$th time-related feature on day $i$. These are, in order, the day of the week, the day in the year, the month, and the season.

We define the corresponding random variables: $S_j^\ell$ is the variable representing the sales at location $\ell$ for product $j$, from which the $p$ individual values $s_{ij}^\ell$, $1 \leq i \leq p$ are assumed to be sampled; $M_j^\ell$ represents the promotion state at $\ell$ for $j$, and similarly for $W_j^\ell$ and $T_j^\ell$.

We further define the substitution of any index $i, j, \ell$ in $s_{ij}^\ell$ by a star $*$ to be the sum over this index of the individual values, e.g.:

$$s_{ij}^* = \sum_{\ell=1}^{m} s_{ij}^\ell \tag{6.1}$$

$$s_{i*}^* = \sum_{\ell=1}^{m} \sum_{j=1}^{k} s_{ij}^\ell, \tag{6.2}$$

whereas substituting any index by a bullet $\bullet$ in $s_{ij}^\ell$ or $w_{ij}^\ell$ and marking it bold denotes the row or column vector (depending on which index is used) formed by the same elements that are summed over above:

$$\mathbf{s}_{\bullet j}^\ell = \left(s_{1j}^\ell, s_{2j}^\ell, \cdots, s_{pj}^\ell\right)^T \tag{6.3}$$

$$\mathbf{w}_{i\bullet}^\ell = \left(w_{i1}^\ell, w_{i2}^\ell, \cdots, w_{iq}^\ell\right). \tag{6.4}$$

In the context of a location $\ell$, we write "the $i$th sample for product $j$" to mean the row vector

$$\mathbf{v}_{ij}^\ell = \left(s_{ij}^\ell, m_{ij}^\ell, \mathbf{w}_{i\bullet}^\ell, \mathbf{t}_{i\bullet}^\ell\right), \tag{6.5}$$

i.e., for day $i$: the transaction value for the product, whether there is an ongoing promotion, and all weather features. We make the *stationarity assumption*: we assume that these samples are i.i.d. according to some joint probability distribution $p\left(\mathbf{V}_j^\ell\right)$ which does not vary with time.

According to a given definition of the working dataset $D$, the graph $\mathcal{G}_D$ is the corresponding causal graph obtained with the PartCorr algorithm, and $\omega_{XY}^{\mathcal{G}_D}$ is the weight of the arc from $X$ to $Y$ in this graph, $0 \leq \omega_{XY} \leq 1$. For undirected edges, we have $\omega_{XY} = \omega_{YX}$; for inexistent links, we have $\omega_{XY} = 0$. The weights $\omega_{XX}$ are always zero.

Categorical variables with more than two possible values have been replaced by a one-hot encoding: a value of the type $X = x_k$ is mapped to a vector $\mathbf{x}$ where all entries are 0 except for the $k$th entry, which is 1.

Note that we simplify the promotion information by summing it up in the binary matrix $\mathbf{M}^\ell$, simply indicating whether or not a promotion is currently running, and not taking into account overlapping promotions. It is straightforward to change its definition to take into account the number of promotions, or to change it into a multinomial variable indicating the form of the ongoing promotion if we also have access to this information. In our tests, the former yielded insignificant changes, while we did not have the required data needed to do the latter.

The final goal is to assess the causal effect of the promotions on the sales, or, alternatively, to find other causes for high or low sales. A lot of products are highly seasonal and we must take care not to be misleaded by it. Other products are known to sell particularly well on certain days of the year; ideally, we would like to answer the question "would the sales have been so high, had we not sent out that special offer a week before?" In order to allow for causal analysis involving other factors, we reserve the possibility to change our definition of sample in Equation 6.5.

The algorithm is meant in this context as a backend engine for a graphical application allowing the end user to select or construct the wanted dataset and run a causality analysis on it. Ideally, the application would allow the user to force certain causal links or link directions and forbid the appearance of others, known not to hold, to be an artifact of the data or to be due to unobserved factors, for instance.

Actually, in many cases, the results are obvious. "Sure, we know that whenever we promote an article, we sell it better," could the salesman say. The intuitive feeling of experts and professional people is usually correct, and in this sense, they might hold little of such a tool. But its power is that it can quantify those relations by producing a graph usable with the *do* calculus.

Another interesting use of it is in the detection of outliers. Detecting promoted products which sell considerably less than other comparable and similarly promoted products could give a hint to the marketing department, which could for instance realize that this is due to the absence of pictures in some flyer for the products yielding lower sales.

In general, we must keep in mind that, working with this data, we do not aim at discovering the true economical or sociological processes going on when customers purchase products, but rather at providing a tool able to analyze thousands of relationships between variables and to point at the outliers. While the human mind has a good understanding of the big picture, it will not be able to examine individually each promotion for each product in each location and detect all special unexpected patterns, while an automatic tool can do it.

## 6.2 EXPERIMENTAL SETUP

The possible scenarios for causal analyses are many. To assess the effect of the promotions on the sales, the simplest scenario is to choose a location $\ell$ and a product $j$ and to run the algorithm on $\mathbf{V}_j^\ell$. We now define a series of experiments of causality analysis (CA) based on this initial idea. In parentheses next to the name of each experiment are parameters needed to fully specify it.

The weight from node PROMOTION to SALES is particularly interesting; we name it $\widetilde{\omega}$ to simplify the notation in some of the following equations.

**Experiment 1** ***Intraproduct, intralocation CA*** $(\ell, j)$

For a particular product in some location, we run PartCorr on the following dataset:

$$D_1 = \left\{ \mathbf{v}_{ij}^\ell \;\middle|\; 1 \leq i \leq p \right\}. \tag{6.6}$$

We then get a causal graph specific to a given product in a given location.

**Experiment 2** ***Interproduct, intralocation CA*** $(\ell)$

We use all products in some location and run PartCorr on the following dataset:

$$D_2 = \bigcup_{j=1}^{k} \{\mathbf{v}_{ij}^{\ell} \mid 1 \leq i \leq p\}. \tag{6.7}$$

We then get a causal graph which can be interpreted as the average causal graph for a given location.

**Variant:** take the union over certain products only, e.g., the $r$ products $\mathbf{j}^*$ with the highest sales:

$$\mathbf{j}^* = \arg\max_{\mathbf{j} \in \mathbb{N}_*^r} \sum_{j \in \mathbf{j}} \sum_{i=1}^{p} s_{ij}^{\ell}, \text{ and} \tag{6.8}$$

$$D_2' = \bigcup_{j \in \mathbf{j}^*} \{\mathbf{v}_{ij}^{\ell} \mid 1 \leq i \leq p\}. \tag{6.9}$$

**Experiment 3** ***Interproduct, intralocation outlier detection*** $(\ell)$

We first run Experiment 1 for all products $j$ and call the resulting graphs $\mathcal{G}_j$. We then run Experiment 2 and call the resulting graph $\mathcal{G}_*$. We now find the most atypical product $j^*$ according to one of the three following definitions:

$$j^* = \arg\max_{j} \sum_{X,Y \in \mathbf{V}} |\omega_{XY}^{\mathcal{G}_j} - \omega_{XY}^{\mathcal{G}_*}| \tag{6.10}$$

$$j^* = \arg\max_{j} |\widetilde{\omega}^{\mathcal{G}_j} - \widetilde{\omega}^{\mathcal{G}_*}| \tag{6.11}$$

$$j^* = \arg\max_{j} \max_{X,Y \in \mathbf{V}} |\omega_{XY}^{\mathcal{G}_j} - \omega_{XY}^{\mathcal{G}_*}|. \tag{6.12}$$

Equation 6.10 says that the most atypical product is the one that differs most from the mean when we sum the differences of arc weights in the whole graph. Equation 6.11 restricts it to the only link between PROMOTION and SALES, while Equation 6.12 looks for the most different arc over the whole graph.

Instead of trying to find this single product $j^*$, we could also rank the products (according to the expression being maximized in the above equations) and sort them according to their ranks: we would then get an ordering of the products from the closest to the most diverging from the average.

**Experiment 4** ***Interproduct, interlocation CA***

We use all products in all locations:

$$D_4 = \bigcup_{\ell=1}^{m} \bigcup_{j=1}^{k} \{\mathbf{v}_{ij}^{\ell} \mid 1 \leq i \leq p\}. \tag{6.13}$$

We then get the most general causal graph we can get, using all available data in the same analysis.

**Experiment 5** ***Intralocation promotion time lag CA*** $(\ell, d_{max})$

For a given location, we run $(d_{max}+1)$ times PartCorr with $0 \leq d \leq d_{max}$ on the following datasets,

modifying the definition of a sample from Equation 6.5:

$$D_5(d) = \bigcup_{j=1}^{k} \left\{ \left( s_{ij}^{\ell}, m_{i-d,j}^{\ell}, \mathbf{w}_{i\bullet}^{\ell}, \mathbf{t}_{i\bullet}^{\ell} \right) \mid 1 \leq i \leq p \right\}. \tag{6.14}$$

We then look for the largest $\widetilde{\omega}$:

$$d^* = \arg \max_{0 \leq d \leq d_{max}} \widetilde{\omega}^{\mathcal{G}(d)}. \tag{6.15}$$

We can thus find the number of days $d^*$ representing the time lag (or lack of time lag if $d^* = 0$) for which the (supposedly) causal effect of the promotion on the sales is maximal.

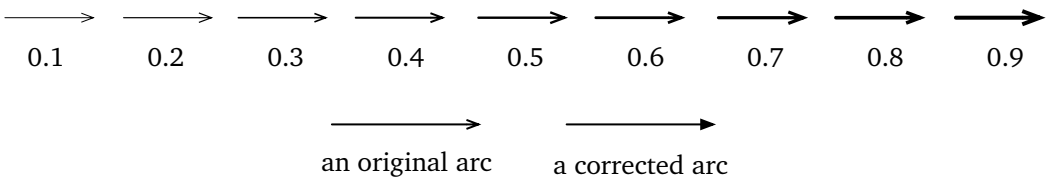
**Variant:** do the same time lag analysis with another variable pair.

Many other kinds of analyses could be conducted. We chose the ones above because we thought they were representative and general enough to be included in this report without disclosing more on details of the marketing campaigns of the involved company.

The next section compares PartCorr with a modified version of the PC algorithm which uses tests of conditional independence using vanishing partial correlations (Scheines *et al.*, 1995) with Fisher's $z$-transform. All experiments were run with Matlab scripts on a 1 Ghz machine.

## 6.3 RESULTS

In the graphs, the line width of an arc from $X$ to $Y$ indicates the weight $\omega_{XY}$ as returned by the algorithm, according to the following mapping:




Some arcs in the graphs were obviously wrongly directed. We know that the number of sunny hours does not influence the season, but most probably the other way around (the other solution being a latent variable). We have thus inverted some arcs by hand, which we draw with a full arrow, as shown above. When it was not obvious that the direction was wrong, it was left as detected by the algorithm.

Generally, the resulting graphs are close to fully-connected between the variables coming from $\mathbf{W}^{\ell}$ and $\mathbf{T}^{\ell}$. As we do not really care about these links and as they will be very similar from product to product and from location to location, we do not show them in each graph. They are shown once for all in Figure 6.1. In further graphs, the nodes MIN. TEMP. and MAX. TEMP. have been merged into a single node TEMPERATURE as the results were more readable and without losing their interpretability.

We first give all graphs—the PartCorr graph in (a) and the PC graph in (b)—and then discuss the results in Section 6.4. In general, PC was one third faster than PartCorr. Although we ran our

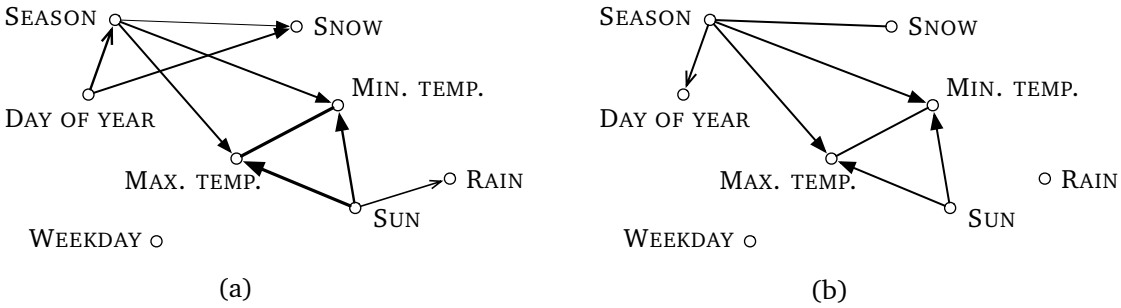



FIG. 6.1: Links between variables from $\mathbf{W}^\ell$ and $\mathbf{T}^\ell$ for $\ell = 1$: (a) PartCorr; (b) PC

experiments in their whole respective parameter spaces, we only present selected results in this document for specific parameter values.

**Experiment 1:** see Figure 6.2.

**Experiment 2:** see Figure 6.3.

**Experiment 3:** see Table 6.1.

Represented in the table are entries of the type "$j : \Delta\omega$" where $j$ is the product where the maximal $\Delta\omega$ (according to the specified formula) was found for a given location. We restricted the analysis to the $r = 10$ products with the highest sales.

**Experiment 4:** see Figure 6.4. We changed the significance level to $\alpha = 0.01$ in PartCorr.

**Experiment 5:** interestingly, for all locations, the optimal time lag was $d^* = 0$ with $d_{max} = 7$.

## 6.4 DISCUSSION

**PartCorr vs. PC**

We first see that PC never outputs a graph identical to the one returned by PartCorr. Why are they different, if both algorithms are proven correct?

First, the correctness proof for PartCorr relies on Hypothesis 5.2.1, which has yet to be proven. Second, we do not know if the problem is DAG-isomorphic. Third, we do not know whether the linearity assumption is reasonable. PartCorr and PC use these assumptions differently. PC removes edges from a complete graph (and thus reduces a trivial I-map to a minimal I-map) and PartCorr adds edges as they are found to exist (and thus augments a trivial D-map to a maximal D-map). PartCorr also conditions on bigger sets when computing partial correlations (and thus has not the same parameters as PC for the statistical test of vanishing partial correlation).

PC systematically outputs sparser graphs than PartCorr in our problem. Often, though not always, the links present according to PartCorr but missing in PC's point of view are drawn thin, and thus correspond to small values of the remaining partial correlation. On the other hand, some links present in PC's output are missing in PartCorr's graphs (3 out of 20 in total). This can also be related to the way PartCorr conditions—using the conditioning set yielding the smallest partial correlation between two nodes $X$ and $Y$. It relies on the DAG-isomorphicity assumption

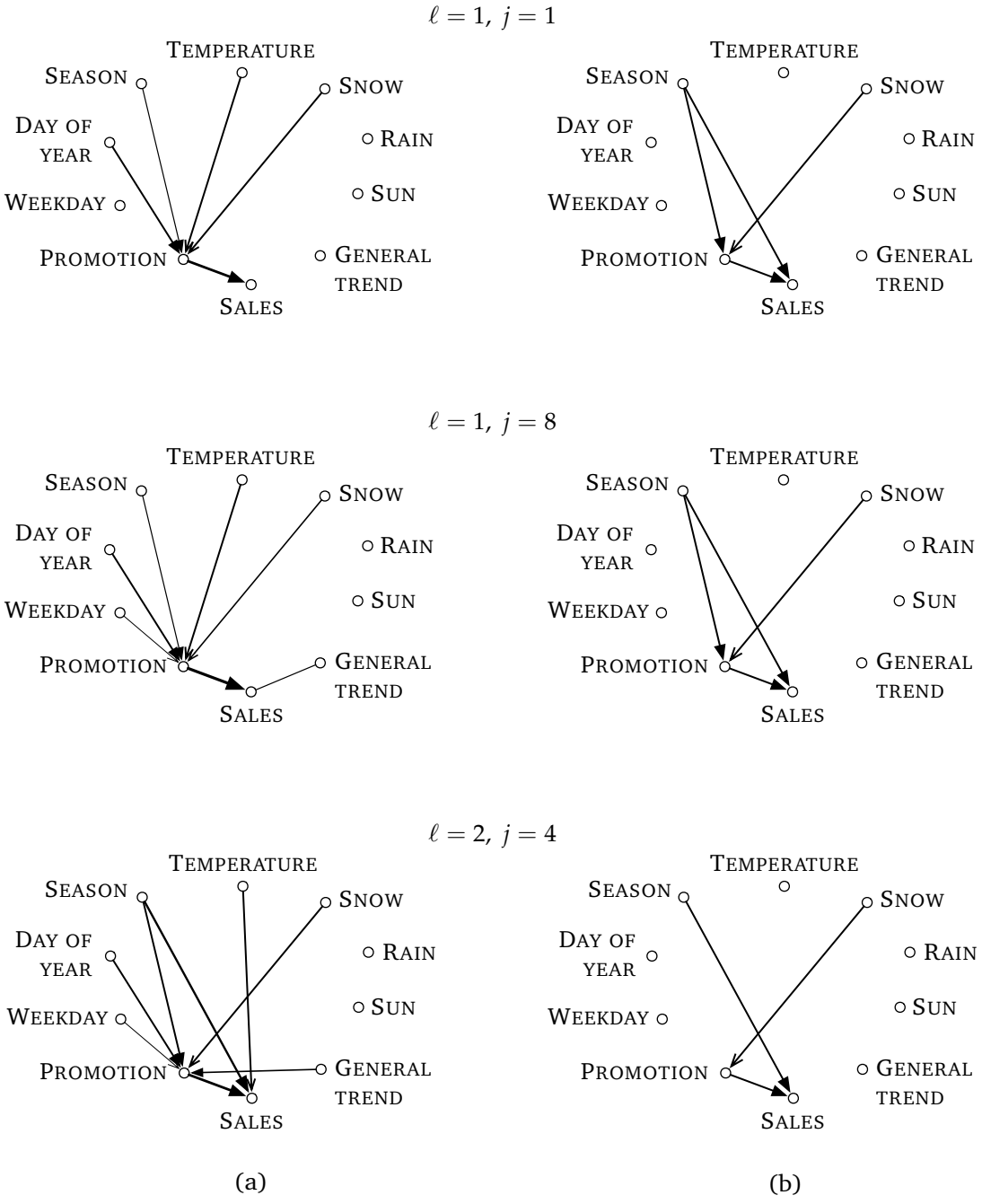



FIG. 6.2: Graphs for Experiment 1: (a) PartCorr; (b) PC

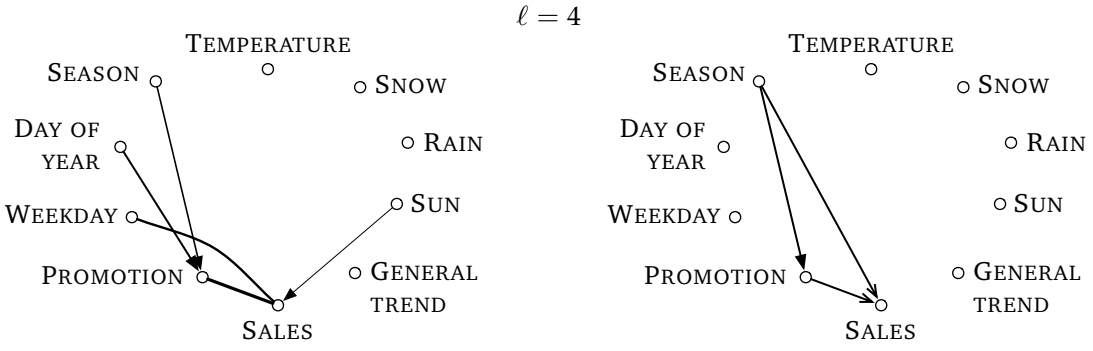



FIG. 6.3: Graphs for Experiment 2: (a) PartCorr; (b) PC

| Location | According to (6.10) | According to (6.11) | According to (6.12) | Which arc |
|---|---|---|---|---|
| $\ell = 1$ | 8 : 2.16 | 6 : 0.18 | 6 : 0.18 | PROMOTION → SALES |
| $\ell = 2$ | 4 : 2.57 | 3 : 0.14 | 1 : 0.19 | SNOW → PROMOTION |
| $\ell = 3$ | 1 : 2.89 | 1 : 0.15 | 8 : 0.31 | TEMPERATURE → PROMOTION |
| $\ell = 4$ | 10 : 2.34 | 7 : 0.17 | 7 : 0.17 | PROMOTION → SALES |
| $\ell = 5$ | 1 : 2.20 | 1 : 0.12 | 7 : 0.20 | DAY OF YEAR → PROMOTION |

TABLE 6.1: Results for Experiment 3.


FIG. 6.4: Graphs for Experiment 4: (a) PartCorr; (b) PC

to ensure that $X$ and $Y$ are $d$-connected in the resulting graph by at least one of the variables in the conditioning set. PC, on the contrary, is perhaps smarter, in that it never conditions on nodes whose links to $X$ or $Y$ have already been discarded (in a previous iteration) and are known to be irrelevant to the conditional independence question.

In many cases, we can extract a meaningful structure skeleton by (conservatively) taking the intersection of the edges of both graphs, or (inclusively) the union. PartCorr's weights also give a hint at the significance of PC's corresponding links, too.

#### Arc Orientation

Both for PC and for PartCorr, the arc orientation has been corrected in many cases. In particular, the arc between PROMOTION and SALES has been reversed 8 times out of 10. This raises the issue of the significance of the edge orientation mechanism in practice, when we are unsure about possible violations of the assumptions—in particular, the absence of causal feedback loops.

Although this is a convenient assumption in theory, it is very limiting in practice. Ideally, we should not exclude that high sales following the promotion of a given article brought the marketing departement to promote the same article again—which is exactly the case of an obvious feedback loop. Of course, causal loops between variables such as SNOW, RAIN and SUN are very likely to exist, and to be entangled with some other hidden, hardly identifiable common cause mechanism.

#### Variance of the Analyses

We see in general that the causal graphs can look more different from each other than expected—and the variance is greater with PartCorr than with PC. This can mean that either the products are indeed different and that customers do not apply the same policy when buying them, or that PartCorr lacks robustness, whereas PC finds more similar patterns.

More interesting investigations into this would be to cluster the products according to the similarity of their causal graphs, and, going back to the original data, find whether the products clustered together are similar in their nature, their price, or the marketing policy applied to them, for instance. Or, conversely, to have the products clustered *a priori* by experts, based on some abstract criteria, and to then determine whether or not the intracluster variance of the resulting causal graphs is smaller than the intercluster variance.

An analysis made possible by PartCorr's arc weights is outlier detection: by looking at a table similar to Table 6.1, it is possible to list products exhibiting a behavior significantly different from the average, and to further investigate into it, either to try to extend this beneficial behavior to other products, or to find the causes for the particularly bad results for this products and get rid of them.

#### Intervening & Confounders

Perhaps the most important information to be read from these causal graphs is the presence or absence of confounders for the variables we will tune in the future. In our case, we can most probably only intervene on the node PROMOTION, and we will do that in a way that will reinforce the sales predictably.

In some graphs, we do detect patterns of the type of the simple example in Figure 2.1 used to illustrate Simpson's paradox. In particular, the nodes SEASON and WEEKDAY are sometimes confounders for PROMOTION and SALES. This means that some products are seasonal (which we know) and that promotions also depend on the season (which we also know). What the causal graph and the confounder pattern underline is that conditioning on, e.g., PROMOTION = *aggressive rebates*, will not accurately describe the effect of this intervention, as explained in Subsection 2.1.4 and detailed in Subsection 3.3.2, but that we have to adjust for the identified confounders to get an unbiased estimate.

This experiment set should be regarded as a first approach to the problem of assessment of the effect of the promotions on the products in the described context: based on these first results, many more experiments can be conducted, especially with more information about the products, product categories, and promotion types.

**Summary**

*Causality analysis with continuous-valued real-world data was conducted with the PartCorr algorithm. Its results are similar but not identical to those of the modified PC algorithm. Several experimental scenarios were described. The structure learning process is not "fool-proof" and the graphs must not be interpreted as depicting The Truth—they only give an idea of the big picture behind the mechanisms generating the data. Outlier detection by looking at variations in the arc weights is a plus allowed by PartCorr.*

CHAPTER

7

# Conclusion & Outlook

ALTHOUGH correlation is not causation in general, we saw that we can define contexts where conditional correlation measures indicate some form of causation. A few basic principles led us to the conclusion that it is feasible to detect at least some cause–effect relationships from observational data.

Looking for a convenient representation of the causal relationships, we motivated the use of directed acyclic graphs together with the $d$-separation criterion to read conditional independencies off the graph. The causal Markov condition and the Faithfulness condition allow us to establish a one-to-one mapping between conditional independence and $d$-separation in a graph; the *do* calculus helps assess the effect of interventions on the system using the structure of the causal graph.

But this all comes at the expense of the assumption that our data is DAG-isomorphic—which excludes a number of problems, like the simple XOR example—and the assumption that no causal feedback loop exists.

With PartCorr, we examined an algorithm which works with continuous variables and gives weights to the causal links. We applied it successfully on real-world retail data to visualize the structure of the underlying processes and to detect outliers by looking at large deviations from the average in the weight of certain arcs.

We also realized that a lot of gaps between the—at last!—emerging causality theory and its application in the real world remain to be filled. Deviations from the DAG-isomorphicity assumption should not make a structure learning algorithm collapse. Ideally, we should be able to detect any kind of functional relationship, and not just the linear ones, with both discrete (or categorical) and continuous variables—within a reasonable computational complexity. We should investigate into the extent to which we can detect and model causal feedback loops and reason over it to predict. Missing values should be allowed without causing data points to be completely ignored.

With this study, we hope to have made the causation detection challenges and their partial solutions a bit more intuitive and accessible by consolidating information from various sources into a single document. We realize that sometimes, in an effort to keep this document readable by the rest of us, details have been skipped, or properties stated without the proper motivation or justification; we hope the interested readers could satisfy their curiosity by looking at the bibliography at the end of this document.

APPENDIX

# A

# Pseudocode of Selected Algorithms

## A.1 GREEDY EQUIVALENCE SEARCH

**Algorithm 2** Greedy Equivalence Search

**procedure** GESCONSTRUCTION
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ empty graph with $n$ nodes

/* Thickening: add edges */
**repeat**
$s_{\mathcal{G}} \leftarrow \text{score}(\mathcal{G}, D)$
$\boldsymbol{\mathcal{E}}^{+} \leftarrow$ PDAGs of all equivalence classes obtained with an edge addition to $\mathcal{G}$
**for each** $\mathcal{G}' \in \boldsymbol{\mathcal{E}}^{+}$ **do** compute $s_{\mathcal{G}'} \leftarrow \text{score}(\mathcal{G}', D)$
**if** $\exists \mathcal{G}' \in \boldsymbol{\mathcal{E}}^{+}$ such that $s_{\mathcal{G}'} > s_{\mathcal{G}}$ **then**
$\mathcal{G} \leftarrow \arg\max_{\mathcal{G}' \in \boldsymbol{\mathcal{E}}^{+}} s_{\mathcal{G}'}$
**end if**
**until** $\forall \mathcal{G}' \in \boldsymbol{\mathcal{E}}^{+} : s_{\mathcal{G}} > s_{\mathcal{G}'}$

/* Thinning: remove edges */
**repeat**
$s_{G} \leftarrow \text{score}(\mathcal{G}, D)$
$\boldsymbol{\mathcal{E}}^{-} \leftarrow$ PDAGs of all equivalence classes obtained with an edge deletion from $\mathcal{G}$
**for each** $\mathcal{G}' \in \boldsymbol{\mathcal{E}}^{-}$ **do** compute $s_{\mathcal{G}'} \leftarrow \text{score}(\mathcal{G}', D)$
**if** $\exists \mathcal{G}' \in \boldsymbol{\mathcal{E}}^{-}$ such that $s_{\mathcal{G}'} > s_{\mathcal{G}}$ **then**
$\mathcal{G} \leftarrow \arg\max_{\mathcal{G}' \in \boldsymbol{\mathcal{E}}^{-}} s_{\mathcal{G}'}$
**end if**
**until** $\forall \mathcal{G}' \in \boldsymbol{\mathcal{E}}^{-} : s_{\mathcal{G}} > s_{\mathcal{G}'}$
**end procedure**

## A.2 PC ALGORITHM

**Algorithm 3** PC Algorithm

**procedure** PCCONSTRUCTION
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ fully connected graph with $n$ nodes
$i \leftarrow 0$

/* Unnecessary arc deletion */
**while** $i \leq$ maximum number of edges for any node **do**
**for each** adjacent unordered pair $X, Y$ such that $|\mathbf{Bd}(X)| > i$ **do**
**if** $\exists$ set $\mathbf{S} \subset \mathbf{Bd}(X)$ of size $i$ such that $(X \perp\!\!\!\perp Y \mid \mathbf{S})$ **then**
remove link $X - Y$ from $\mathcal{G}$
$\mathbf{S}_{XY}, \mathbf{S}_{YX} \leftarrow \mathbf{S}$
**end if**
**end for**
$i \leftarrow i + 1$
**end while**

/* V-structure detection */
**for each** $X, Y, Z$ such that $X - Z - Y$ **do**
**if** $Z \notin \mathbf{S}_{XY}$ **then** orient as $X \rightarrow Z \leftarrow Y$
**end for**

/* Constraint propagation */
**while** $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
**for each** $X, Y, Z$ such that $X \rightarrow Y - Z$ **do**
orient as $X \rightarrow Y \rightarrow Z$ /* no new V-structure */
**end for**
**for each** $X, Y$ such that $X - Y$ and $\exists$ directed path from $X$ to $Y$ **do**
orient as $X \rightarrow Y$ /* preserve acyclicity */
**end for**
**for each** $X, Y$ s.t. $X - Y$ and $\exists$ nonadjacent $Z, W$ s.t. $X - Z \rightarrow Y$ and $X - W \rightarrow Y$ **do**
orient as $X \rightarrow Y$ /* three-fork V with married parents */
**end for**
**end while**
**end procedure**

## A.3 IC* ALGORITHM

**Notation**

| | |
|---|---|
| $X \ast\!\!\rightarrow Y$ | Any directed, bidirected, marked arc that has an arrow pointing into $Y$; |
| $X \ast\!\!- Y$ | Any edge that does not have an arrow pointing into $Y$. |

**Algorithm 4** IC* Algorithm

**procedure** ICSTARCONSTRUCTION
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ fully connected graph with $n$ nodes
$i \leftarrow 0$

/* Unnecessary arc deletion */
**while** $i \leq$ maximum number of edges for any node **do**
  **for each** adjacent unordered pair $X, Y$ such that $|\mathbf{Bd}(X)| > i$ **do**
    **if** $\exists$ set $\mathbf{S} \subset \mathbf{Bd}(X)$ of size $i$ such that $(X \perp\!\!\!\perp Y \mid \mathbf{S})$ **then**
      remove link $X - Y$ from $\mathcal{G}$
      $\mathbf{S}_{XY}, \mathbf{S}_{YX} \leftarrow \mathbf{S}$
    **end if**
  **end for**
  $i \leftarrow i + 1$
**end while**

/* V-structure detection */
**for each** $X, Y, Z$ such that $X \ast\!\!- Z \ast\!\!\rightarrow Y$ **do**
  **if** $Z \notin \mathbf{S}_{XY}$ **then** orient as $X \ast\!\!\rightarrow Z \leftarrow\!\!\ast Y$
**end for**

/* Constraint propagation */
**while** $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
  **for each** $X, Y, Z$ such that $X \ast\!\!\rightarrow Y - Z$ **do**
    orient as $X \ast\!\!\rightarrow Y \overset{*}{\rightarrow} Z$ /* no new V-structure */
  **end for**
  **for each** $X, Y$ such that $X \ast\!\!- Y$ and $\exists$ marked directed path from $X$ to $Y$ **do**
    orient as $X \ast\!\!\rightarrow Y$ /* preserve acyclicity */
  **end for**
**end while**
**end procedure**

## A.4 CBL ALGORITHM

Another version of the CBL algorithm does not require a prior ordering to orient the causal links, at the expense of more conditional independence tests. This is the original version.

**Algorithm 5** CBL Algorithm

**procedure** CBLCONSTRUCTION
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**o** : topological order of the variables with respect to the causal graph
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ empty graph with $n$ nodes
$\mathbf{R} \leftarrow \emptyset$
**for each** undirected pair $(X, Y)$ **do**
$m_{XY} \leftarrow I(X; Y)$ /* pairwise mutual information */
**end for**

/* Drafting */
**for each** undirected pair $(X, Y)$ such that $m_{XY}$ is significant **do**
**if** $\text{dSep}(X, Y \mid \emptyset)_{\mathcal{G}}$ **then**
add an arc between $X$ and $Y$ to $\mathcal{G}$, directing it according to **o**
**else**
$\mathbf{R} \leftarrow \mathbf{R} \cup \{(X, Y)\}$
**end if**
**end for**

/* Thickening */
**for each** undirected pair $(X, Y) \in \mathbf{R}$ **do**
$\mathbf{S}_{XY} \leftarrow$ the minimum set $d$-separating $X$ from $Y$ in $\mathcal{G}$
**if** $(X \not\perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$ **then**
add an arc between $X$ and $Y$ to $\mathcal{G}$, directing it according to **o**
**end if**
**end for**

/* Thinning */
**for each** arc $X \rightarrow Y$ in $\mathcal{G}$ **do**
$\mathcal{G}_{\overline{X \rightarrow Y}} \leftarrow$ the graph where the arc from $X$ to $Y$ has been removed
$\mathbf{S}_{XY} \leftarrow$ the minimum set $d$-separating $X$ from $Y$ in $\mathcal{G}_{\overline{X \rightarrow Y}}$
**if** $(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$ **then**
remove the arc $X \rightarrow Y$ from $\mathcal{G}$
**end if**
**end for**
**end procedure**

## A.5 PARTCORR ALGORITHM

**Algorithm 6** PartCorr construction algorithm

**procedure** PARTCORRCONSTR
**Input:** $D$ : $p \times n$ dataset with $p$ $n$-dimensional data points
**Output:** $\mathcal{G}$ : partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ empty graph with $n$ nodes
normalize $D$ so that variables have a zero mean
**for each** undirected pair $(X, Y)$ **do** $coll_{XY} \leftarrow \emptyset$

/* Undirected structure construction */
**for each** undirected pair $(X, Y)$ **do**
$coll_{XY} \leftarrow$ the set of all nodes $Z \neq X, Y$ such that $|\rho_{XY \cdot Z}| > |\rho_{XY}|$
$\mathbf{S}_{XY} \leftarrow \mathbf{V} \setminus coll_{XY} \setminus \{X, Y\}$
**if** $\rho_{XY \cdot \mathbf{S}_{XY}}$ does not vanish **then**
add an undirected edge $X - Y$ to $\mathcal{G}$
**end if**
**end for**

/* V-structure detection */
**for each** $X, Y, Z$ such that $X - Z - Y$ **do**
**if** $Z \notin \mathbf{S}_{XY}$ **then** orient as $X \rightarrow Z \leftarrow Y$
**end for**

/* Constraint propagation */
**while** $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
**for each** $X, Y, Z$ such that $X \rightarrow Y - Z$ **do**
orient as $X \rightarrow Y \rightarrow Z$ /* no new V-structure */
**end for**
**for each** $X, Y$ such that $X - Y$ and $\exists$ directed path from $X$ to $Y$ **do**
orient as $X \rightarrow Y$ /* preserve acyclicity */
**end for**
**for each** $X, Y$ s.t. $X - Y$ and $\exists$ nonadjacent $Z, W$ s.t. $X - Z \rightarrow Y$ and $X - W \rightarrow Y$ **do**
orient as $X \rightarrow Y$ /* three-fork V with married parents */
**end for**
**end while**
**end procedure**

APPENDIX

**B**

# Mathematical Proofs

## B.1 First-Order Partial Correlation

We show that the partial correlation $\rho_{XY \cdot Z}$ can be expressed in terms of the zero-order correlations $\rho_{XY}$, $\rho_{YZ}$ and $\rho_{XZ}$ according to the following formula:

$$\rho_{XY \cdot Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{YZ}}{\sqrt{1-\rho_{XZ}^2}\sqrt{1-\rho_{YZ}^2}}$$

For readability, we use interchangeably the notations $\rho_{XY}$ and $\text{corr}(X, Y)$ to denote Pearson's correlation. The partial correlation $\rho_{XY \cdot Z}$ is then defined as the correlation of the residuals $R_X$ and $R_Y$ resulting from a linear regression on $X$ with $Z$ and on $Y$ with $Z$, respectively. Without loss of generality, we assume centered variables: $\mathbb{E}(X) = \mathbb{E}(Y) = \mathbb{E}(Z) = 0$.

$$\begin{aligned}
r_X &= a^* Z - X \\
a^* &= \arg\min_a \mathbb{E}\left[(aZ - X)^2\right] \\
0 &= \mathbb{E}[2(a^* Z - X)Z] \\
a^* &= \frac{\mathbb{E}(XZ)}{\mathbb{E}(Z^2)}
\end{aligned}$$

We can now write the correlation of the residuals $R_X = \frac{\mathbb{E}(XZ)}{\mathbb{E}(Z^2)}Z - X$, $R_Y = \frac{\mathbb{E}(YZ)}{\mathbb{E}(Z^2)}Z - Y$. We have:

$$\text{corr}(R_X, R_Y) = \frac{\text{cov}(R_X, R_Y)}{\sqrt{\text{var}(R_X)}\sqrt{\text{var}(R_Y)}},$$

so let's compute the covariance and the variances.

$$\text{cov}(R_X, R_Y) = \mathbb{E}(R_X R_Y) - \mathbb{E}(R_X)\,\mathbb{E}(R_Y)$$

$$\begin{aligned}
\mathbb{E}(R_X R_Y) &= \mathbb{E}\left[\left(\frac{\mathbb{E}(XZ)}{\mathbb{E}(Z^2)}Z - X\right)\left(\frac{\mathbb{E}(YZ)}{\mathbb{E}(Z^2)}Z - Y\right)\right] \\
&= \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\mathbb{E}(Z^2)^2}\,\mathbb{E}(Z^2) - \frac{\mathbb{E}(XY)\,\mathbb{E}(XZ)}{\mathbb{E}(Z^2)} - \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\mathbb{E}(Z^2)} + \mathbb{E}(ZY) \\
&= \mathbb{E}(XY) - \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\mathbb{E}(Z^2)} \\
\mathbb{E}(R_X)\,\mathbb{E}(R_Y) &= \left(\frac{\mathbb{E}(XZ)}{\mathbb{E}(Z^2)}\underbrace{\mathbb{E}(Z)}_{0} - \underbrace{\mathbb{E}(X)}_{0}\right)\left(\frac{\mathbb{E}(YZ)}{\mathbb{E}(Z^2)}\underbrace{\mathbb{E}(Z)}_{0} - \underbrace{\mathbb{E}(Y)}_{0}\right) \\
&= 0 \cdot 0 = 0 \\
\operatorname{cov}(R_X, R_Y) &= \mathbb{E}(XY) - \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\mathbb{E}(Z^2)}.
\end{aligned}$$

We now have the numerator of the wanted correlation. Let's now turn to the denominator:

$$\begin{aligned}
\operatorname{var}(R_X) &= \mathbb{E}(R_X^2) - \underbrace{\mathbb{E}(R_X)^2}_{0} \\
\mathbb{E}(R_X^2) &= \mathbb{E}\left[\left(\frac{\mathbb{E}(XZ)}{\mathbb{E}(Z^2)}Z - X\right)^2\right] \\
&= \frac{\mathbb{E}(XZ)^2}{\mathbb{E}(Z^2)^2}\,\mathbb{E}(Z^2) - 2\frac{\mathbb{E}(XZ)^2}{\mathbb{E}(Z^2)} + \mathbb{E}(X^2) \\
&= \mathbb{E}(X^2) - \frac{\mathbb{E}(XZ)^2}{\mathbb{E}(Z^2)}.
\end{aligned}$$

Putting numerator and denominator together and dividing both (second line) by $\sqrt{\mathbb{E}(X^2)}\sqrt{\mathbb{E}(Y^2)}$, we have:

$$\begin{aligned}
\operatorname{corr}(R_X, R_Y) &= \frac{\mathbb{E}(XY) - \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\mathbb{E}(Z^2)}}{\sqrt{\mathbb{E}(X^2) - \frac{\mathbb{E}(XZ)^2}{\mathbb{E}(Z^2)}}\sqrt{\mathbb{E}(Y^2) - \frac{\mathbb{E}(YZ)^2}{\mathbb{E}(Z^2)}}} \\
&= \frac{\frac{\mathbb{E}(XY)}{\sqrt{\mathbb{E}(X^2)}\sqrt{\mathbb{E}(Y^2)}} - \frac{\mathbb{E}(XZ)\,\mathbb{E}(YZ)}{\sqrt{\mathbb{E}(X^2)}\sqrt{\mathbb{E}(Y^2)}\,\mathbb{E}(Z^2)}}{\sqrt{1 - \frac{\mathbb{E}(XZ)^2}{\mathbb{E}(Z^2)\,\mathbb{E}(X^2)}}\sqrt{1 - \frac{\mathbb{E}(YZ)^2}{\mathbb{E}(Z^2)\,\mathbb{E}(Y^2)}}} \\
&= \frac{\operatorname{corr}(X,Y) - \operatorname{corr}(X,Z)\operatorname{corr}(Y,Z)}{\sqrt{1 - \operatorname{corr}(X,Z)^2}\sqrt{1 - \operatorname{corr}(Y,Z)^2}},
\end{aligned}$$

which is the wanted partial correlation in terms of the zero-order correlation.

## B.2 KERNELIZED LEAST SQUARES METHOD

### B.2.1 Linear Regression

Bold lowercase variables are column vectors, roman lowercase variables are scalars, uppercase variables are matrices (and not random variables). The matrix $\mathbf{X}$ is the $p \times n$ regressor matrix formed by $p$ $n$-dimensional i.i.d. samples: $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \end{pmatrix}^T$.

We have $p$ samples. For each sample $i$, we have $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$; generally, $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^p$. We have a linear regression problem of the form

$$\arg\min_{\mathbf{w}} f(\mathbf{w})$$
$$f(\mathbf{w}) = \sum_{i=1}^{p} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2.$$

Using the matrix notation, this is equivalent to

$$\begin{aligned} f(\mathbf{w}) &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \\ &= (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y}). \end{aligned}$$

Taking the derivative with respect to $\mathbf{w}$ yields

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{w}} &= 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) = 0 \\ \mathbf{X}^T\mathbf{X}\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \mathbf{w} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \end{aligned}$$

Defining $\mathbf{w} = \mathbf{X}^T\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} \in \mathbb{R}^n$) and (second line) left-multiplying by $\mathbf{X}$, we have:

$$\begin{aligned} \mathbf{X}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} &= \mathbf{X}^T\mathbf{y} \\ \mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} &= \mathbf{X}\mathbf{X}^T\mathbf{y}. \end{aligned}$$

Now, let $\mathbf{K} = \mathbf{X}\mathbf{X}^T$:

$$\begin{aligned} \mathbf{K}^2\boldsymbol{\alpha} &= \mathbf{K}\mathbf{y} \\ \boldsymbol{\alpha} &= \mathbf{K}^{-1}\mathbf{y}. \end{aligned}$$

The optimal weight vector $\mathbf{w}^*$ is then

$$\mathbf{w}^* = \mathbf{X}^T\mathbf{K}^{-1}\mathbf{y}.$$

### B.2.2 Ridge Regression

Restating the regression problem to include a ridge, we now have:

$$\begin{aligned} f(\mathbf{w}) &= \sum_{i=1}^{n}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2 + \lambda\|\mathbf{w}\|^2 \\ &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2 \\ &= (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w}^T\mathbf{w} \\ \frac{\partial f}{\partial \mathbf{w}} &= 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) - 2\lambda\mathbf{w} \\ &= 2(\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y} + \lambda\mathbf{w}) = 0 \\ (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \mathbf{w} &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \end{aligned}$$

Like before, with $\mathbf{w} = \mathbf{X}^T\boldsymbol{\alpha}$ and $\mathbf{K} = \mathbf{X}\mathbf{X}^T$:

$$\begin{aligned} (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^T\boldsymbol{\alpha} &= \mathbf{X}^T\mathbf{y} \\ (\mathbf{X}^T\mathbf{X}\mathbf{X}^T + \lambda\mathbf{X}^T)\boldsymbol{\alpha} &= \mathbf{X}^T\mathbf{y} \\ (\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{X}^T + \lambda\mathbf{X}\mathbf{X}^T)\boldsymbol{\alpha} &= \mathbf{X}\mathbf{X}^T\mathbf{y} \\ (\mathbf{K}^2 + \lambda\mathbf{K})\boldsymbol{\alpha} &= \mathbf{K}\mathbf{y} \\ \boldsymbol{\alpha} &= (\mathbf{K}^2 + \lambda\mathbf{K})^{-1}\mathbf{K}\mathbf{y}. \end{aligned}$$

The optimal weight vector $\mathbf{w}^*$ is now

$$\mathbf{w}^* = \mathbf{X}^T(\mathbf{K}^2 + \lambda\mathbf{K})^{-1}\mathbf{K}\mathbf{y}.$$

# Bibliography


ACID, S., & DE CAMPOS, L. M. 1996. An Algorithm for Finding Minimum d-Separating Sets in Belief Networks. *In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*. Available from: citeseer.ist.psu.edu/article/acid96algorithm.html.

AIZERMAN, M. A., BRAVERMAN, E. M., , & ROZONOER, L. I. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, **25**, 821–837.

ALIFERIS, C. F., TSAMARDINOS, I., & STATNIKOV, A. 2003. HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection. *Pages 21–25 of: Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*. Available from: citeseer.ist.psu.edu/aliferis03hiton.html.

BABA, K., SHIBATA, R., & SIBUYA, M. 2004. Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, **46**(4).

BAKALAR, N. 2006, August 15th. Coffee as a health drink? Studies find some benefits. *The New York Times*.

BEINLICH, I., SUERMONDT, H. J., CHAVEZ, R. M., & COOPER, G. F. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Pages 247–256 of: Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, vol. 38.

BERGSMA, W. 2004. Testing conditional independence for continuous random variables. *Journal of the American Statistical Association*. Available from: eprints.pascal-network.org/archive/00000824/01/partial-assoc_new2.pdf.

CARPENTER, J., & KENWARD, M. 2006. *Missing Data*. Website. www.lshtm.ac.uk/msu/missingdata/index.html.

CHARNIAK, E. 1991. Bayesian networks without tears. *The AI Magazine*, **12**(4), 50–63. Available from: www.cs.dartmouth.edu/~donaldclass/GradAI/Readings/BayesNet_Charniak_Presentation.pdf.

CHENG, J., & BELL, D. 1997. Learning Bayesian networks from data: an efficient approach based on information theory. *In: Proceeding of the 6th ACM International Conference on Information and Knowledge Management*. Available from: citeseer.ist.psu.edu/134587.html.

CHENG, J., BELL, D., & LIU, W. 1997a. An algorithm for Bayesian belief network construction from data. *Pages 83–90 of: Proceedings of AI & STAT'97*. Florida: Ft. Lauderdale. Available from: citeseer.ist.psu.edu/cheng97algorithm.html.

CHENG, J., BELL, D., & LIU, W. 1997b. Learning belief networks from data: An information theory based approach. *In: Proceedings of ACM Conference on Information and Knowledge Management '97*. Available from: citeseer.ist.psu.edu/56468.html.

CHICKERING, D. M. 2002. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, **3**, 507–554. Available from: jmlr.csail.mit.edu/papers/volume3/chickering02b/chickering02b.pdf.

CHOW, C. K., & LIU, C. N. 1968. Approximating discrete probability distribution with dependence trees. *IEEE Transactions on Information Theory*, **14**(3), 462–467.

COOPER, G. F., & HERSKOVITS, E. H. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–347.

DEVROYE, L., GYORFI, L., KRZYZAK, A., & LUGOSI, G. 1994. The strong universal consistency of nearest neighbor regression function estimates. *Annals of Statistics*, **22**, 1371–1385. Available from: citeseer.ist.psu.edu/devroye92strong.html.

EINSTEIN, A. 1953. Letter to J.S. Switzer. *Einstein Archive*.

ELGAMMAL, A., DURAISWAMI, R., & DAVIS, L. S. 2003. Efficient Kernel Density Estimation Using the Fast Gauss Transform With Applications to Color Modeling and Tracing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(11). Available from: citeseer.ist.psu.edu/elgammal03efficient.html.

ELIDAN, G., LOTNER, N., FRIEDMAN, N., & KOLLER, D. 2000. Discovering Hidden Variables: A Structure-Based Approach. *Pages 479–485 of: Proceedings of the 13th Conference on Advances in Neural Information Processing Systems*. Available from: citeseer.ist.psu.edu/elidan01discovering.html.

FASER, A.M., & SWINNEY, H.L. 1986. Independent coordinates for strange attractors from mutual information. *Physical Review A*, **33**, 2318–2321.

FLEURET, F. 2004. Fast Binary Feature Selection with Conditional Mutual Information. *Journal of Machine Learning Research*, **5**, 1531–1555. Available from: cvlab.epfl.ch/~fleuret/papers/fleuret-jmlr2004.pdf.

FORD, L.R. (JR.), & FULKERSON, D.R. 1956. Maximal Flow Through a Network. *Canadian Journal of Mathematics*, **8**, 99–404.

FRIEDMAN, N. 1997. Learning belief networks in the presence of missing values and hidden variables. *In: Proceedings of the 14th International Conference on Machine Learning*. Available from: www.cs.huji.ac.il/~nir/Papers/Fr1.pdf.

FRIEDMAN, N. 1998. The Bayesian structural EM algorithm. *Pages 129–138 of: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. Available from: citeseer.ist.psu.edu/article/friedman98bayesian.html.

FRIEDMAN, N., GOLDSZMIDT, M., & WYNER, A. 1999. Data Analysis with Bayesian Networks: A Bootstrap Approach. *Pages 196–205 of: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*. Available from: citeseer.ist.psu.edu/friedman99data.html.

GALLES, D., & PEARL, J. 1997. Axioms of Causal Relevance. *Artificial Intelligence*, **97**(1-2), 9–43. Available from: citeseer.ist.psu.edu/galles96axioms.html.

GRAEDON, J., & GRAEDON, T. 2006, August 14th. A valuable drug, with side effects. *The Los Angeles Times*.

GRANGER, C. W. J. 1988. Some Recent Developments in a Concept of Causality. *Journal of Econometrics*, **39**(1), 199–211.

GRANGER, C. W. J., & ENGLE, R. F. 1987. Co-integration and Error Correction: Representation, Estimation and Testing. *Econometrica*, **55**, 251–276.

GREENGARD, L., & STRAIN, J. 1991. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, **12**(1), 79–94.

GUYON, I., & ELISSEEFF, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157–1182. Available from: jmlr.csail.mit.edu/papers/volume3/guyon03a/guyon03a.pdf.

GUYON, I., WESTON, J., BARNHILL, S., & VAPNIK, V. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, **46**(1-3), 389–422. Available from: citeseer.ist.psu.edu/guyon02gene.html.

HAGMAYER, Y., SLOMAN, S. A., LAGNADO, D. A., & WALDMANN, M. R. 2005. *Causal Reasoning through Intervention*. Tech. rept. University of Goettingen, Brown University, University College London. Available from: else.econ.ucl.ac.uk/papers/uploaded/199.pdf.

HASTINGS, W.K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

HAUSMAN, D. M., & WOODWARD, J. 1999. Independence, Invariance and the Causal Markov Condition. *British Journal for the Philosophy of Science*, **50**, 521–583.

HERSKOVITS, E. H. 1991. *Computer-based probabilistic-network construction*. Ph.D. thesis, Stanford University.

HOERL, A. E., & KENNARD, R. W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(3), 55–67.

IHLER, A. 2004. *Kernel Density Estimation Toolbox for Matlab*. Available from: www.ics.uci.edu/~ihler/code/kde.shtml.

KOHAVI, R., & JOHN, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence*, **97**, 273–324. Available from: citeseer.ist.psu.edu/13663.html.

KRUSKAL, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Pages 48–50 of: Proceedings of the American Mathematical Society*, vol. 7.

LAURITZEN, S. L., DAWID, A. P., LARSEN, B. N., & LEIMER, H.-G. 1990. Independence properties of directed Markov fields. *Networks*, **20**, 491–505.

LERAY, P., & FRANÇOIS, O. 2004a. *BNT Structure Learning Package*. Available from: banquiseasi.insa-rouen.fr/projects/bnt-slp/.

LERAY, P., & FRANÇOIS, O. 2004b. *BNT Structure Learning Package: documentation and experiments*. Tech. rept. INSA Rouen, France. Available from: bnt.insa-rouen.fr/programmes/BNT_StructureLearning_v1.3.pdf.

LERAY, P., & FRANÇOIS, O. 2005. Bayesian Network Structural Learning and Incomplete Data. *Pages 33–40 of: Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Available from: www.cis.hut.fi/AKRR05/papers/akrr05leray.pdf.

MARGARITIS, D. 2005. Distribution-free learning of Bayesian network structure in continuous domains. *In: Proceedings of the 20th National Conference on Artificial Intelligence*. Available from: www.cs.iastate.edu/~dmarg/Papers/Margaritis-AAAI05.pdf.

MARGARITIS, D., & THRUN, S. 2001. A Bayesian Multiresolution Independence Test for Continuous Variables. *Pages 346–353 of: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann. Available from: www.cs.iastate.edu/~dmarg/Papers/UAI-2001.ps.

MEEK, C. 1995. Causal inference and causal explanation with background knowledge. *In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*.

MEEK, C. 1997. *Graphical Models: Selecting causal and statistical models*. Ph.D. thesis, Carnegie Mellon University.

MEILA, M., & JORDAN, M. I. 1998. Estimating dependency structure as a hidden variable. *In: Proceedings of the 10th Conference on Advances in Neural Information Processing Systems*. Available from: citeseer.ist.psu.edu/meila98estimating.html.

MERCER, J. 1909. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, **A 209**, 415–446.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., & TELLER, E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092. Available from: link.aip.org/link/JCPSA6/v21/p1087.

MOON, Y. I., RAJAGOPALAN, B., & LALL, U. 1995. Estimation of mutual information using kernel density estimators. *Physical Review E*, **52**, 2318–2321.

MOUSSOURIS, J. 1974. Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, **10**, 11–33.

MURPHY, K. 2000. *Bayes Net Toolbox for Matlab*. Available from: bnt.sourceforge.net.

MYERS, J. W., LASKEY, K. B., & DEJONG, K. A. 1999. Learning Bayesian Networks from Incomplete Data using Evolutionary Algorithms. *Pages 458–465 of: Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1. Orlando, Florida, USA: Morgan Kaufmann. Available from: citeseer.ist.psu.edu/article/myers99learning.html.

NEAPOLITAN, R. E. 2003. *Learning Bayesian Networks*. Prentice Hall.

O'CONNOR, A. 2006, August 15th. Deodorizers may cause reduction in lung function. *The Chicago Tribune*.

PEARL, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Los Altos: Morgan Kaufmann.

PEARL, J. 1995. Causal diagrams for empirical research. *Biometrika*, **82**(4), 669–709. Available from: citeseer.ist.psu.edu/55450.html.

PEARL, J. 2000. *Causality: Models, Reasoning and Inference*. Cambridge University Press. Available from: singapore.cs.ucla.edu/BOOK-2K/.

PEARL, J., & VERMA, T. 1991. A theory of inferred causation. *In: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. San Mateo, CA: Morgan Kaufmann. Available from: citeseer.ist.psu.edu/pearl91theory.html.

PRIM, R. C. 1957. Shortest connection networks and some generalizations. *Bell System Technical Journal*, **36**, 1389–1401.

QUINE, W. V. O. 1954. The Scope of Language and Thought. *The Ways of Paradox and Other Essays (1966)*.

RICHARDSON, T., & SPIRTES, P. 2002. Ancestral graph Markov models. *Annals of Statistics*, **59**, 845–862. Available from: www.stat.washington.edu/www/research/reports/2000/tr375.ps.

RIGGELSEN, C. 2006. Learning Bayesian networks from incomplete data: an efficient method for generating approximate predictive distributions. *In: SIAM Conference on Data Mining*. Available from: www.siam.org/meetings/sdm06/proceedings/012riggelsenc.pdf.

ROBINSON, R. W. 1977. Counting unlabeled acyclic digraphs. *Combinatorial Mathematics, Lecture Notes in Mathematics*, **622**, 28–43.

RUMMEL, R. J. 1976. *Understanding Correlation*. Website. www.mega.nu/ampp/rummel/uc.htm.

RUSSELL, B. 1913. On the Notion of Cause. *Pages 1–26 of: Proceedings of the Aristotelian Society*, vol. 13.

SCHEINES, R., SPIRTES, P., GLYMOUR, C., MEEK, C., & RICHARDSON, T. 1995. *The TETRAD Project: Constraint Based Aids to Causal Model Specification*. Tech. rept. Carnegie Mellon University, Dpt. of Philosophy. Available from: citeseer.ist.psu.edu/288318.html.

SCHÄFER, J., & STRIMMER, K. 2005. Learning large-scale graphical Gaussian models from genomic data. *Pages 263–276 of:* MENDES, J. F. F., DOROGOVTSEV, S. N., POVOLOTSKY, A., ABREU, F. V., & OLIVEIRA, J. G. (eds), *AIP Conference Proceedings 776*. Available from: stat.ethz.ch/~schaefer/publications/ggm-review2005.pdf.

SHACHTER, R. 1998. Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *Pages 480–487 of:* COOPER, G. F., & MORAL, S. (eds), *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann. Available from: citeseer.ist.psu.edu/shachter98bayesball.html.

SILVA, R., SCHEINES, R., GLYMOUR, C., & SPIRTES, P. 2006. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, **7**, 191–246. Available from: www.gatsby.ucl.ac.uk/~rbas/papers/jmlr-final.pdf.

SILVERMAN, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

SIMPSON, E. H. 1951. The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society, Ser. B*, **13**, 238–241.

SINGH, M. 1997. Learning Bayesian networks from incomplete data. *Pages 27–31 of: Proceedings of the National Conference on Artificial Intelligence*. AAAI Press.

SMOLA, A. J., & SCHÖLKOPF, B. 1998. *A tutorial on support vector regression*. Tech. rept. NeuroCOLT2. Available from: citeseer.ist.psu.edu/smola98tutorial.html.

SPIRTES, P., GLYMOUR, C., & SCHEINES, R. 1993. *Causation, Prediction, and Search*. Vol. 81. Berlin: Springer Verlag.

SPIRTES, P., MEEK, C., & RICHARDSON, T. 1995. Causal Inference in the Presence of Latent Variables and Selection Bias. *Pages 491–498 of:* BESNARD, PHILIPPE, & HANKS, STEVE (eds), *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann. Available from: citeseer.ist.psu.edu/49478.html.

STEEL, D. 2005. *Homogeneity, Selection, and the Faithfulness Condition*. Tech. rept. Michigan State University, Department of Philosophy. Available from: www.msu.edu/user/steel/FCM&M.pdf.

STEUER, R., KURTHS, J., DAUB, C.O., WEISE, J., & SELBIG, J. 2002. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics*, **18 Suppl. 2**, S231–S240. Available from: bioinformatics.oxfordjournals.org/cgi/reprint/18/suppl_2/S231.pdf.

THIESSON, B. 1995. Accelerated Quantification of Bayesian Networks with Incomplete Data. *Pages 306–311 of: Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*. Available from: citeseer.ist.psu.edu/thiesson95accelerated.html.

TSAMARDINOS, I., ALIFERIS, C. F., & STATNIKOV, A. 2003. Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. *Pages 673–678 of:* PRESS, ACM (ed), *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Available from: citeseer.ist.psu.edu/tsamardinos03time.html.

TSAMARDINOS, I., BROWN, L. E., & ALIFERIS, C. F. 2006. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*. Available from: discover1.mc.vanderbilt.edu/discover/public/Publications/DSL-05-01_MergedUpdate2.pdf.

VALORTA, M., & HUANG, Y. 2006a. Identifiability in Causal Bayesian Networks: A Sound and Complete Algorithm. *Pages 1149–1154 of: Proceedings of the 21st National Conference on Artificial Intelligence*. Available from: www.cse.sc.edu/~mgv/papers/HuangValtortaAAAI06.pdf.

VALORTA, M., & HUANG, Y. 2006b. Pearl's Calculus of Intervention is Complete. *Pages 437–444 of: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*. Available from: www.cse.sc.edu/~mgv/papers/HuangValtortaUAI06.pdf.

WESTON, J., ELISSEEFF, A., BAKIR, G., & SINZ, F. 2003. *The Spider machine learning library*. Available from: www.kyb.mpg.de/bs/people/spider/main.html.

WIDDOWS, D. 2004. *Geometry and Meaning*. CSLI Publications. Chapter 4. Available from: infomap.stanford.edu/book/chapters/chapter4.html.

WONG, S. K. M., WU, D., & LIN, T. 2002. A Structural Characterization of DAG-Isomorphic Dependency Models. *Pages 195–209 of:* COHEN, R., & SPENCER, B. (eds), *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence*. Calgary, Canada: Morgan Kaufmann.

WOODWARD, J. 2003. *Making things happen. A theory of causal explanation*. Oxford University Press.

WRIGHT, S. 1923. The theory of path coefficients: A reply to Niles' criticism. *Genetics*, **8**.

YANG, C., DURAISWAMI, R., GUMEROV, N. A., & DAVIS, L. 2003. Improved fast gauss transform and efficient kernel density estimation. *Pages 664–671 of: Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 1. Available from: citeseer.ist.psu.edu/yang03improved.html.

YULE, G. U. 1903. Notes on the theory of association of attributes in Statistics. *Biometrika*, **2**, 121–134.