Causal Discovery
Beyond Conditional Independences

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Eleni Sgouritsa
aus Athens/Greece

Tübingen
2015

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	29 October 2015
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Bernhard Schölkopf
2. Berichterstatter:	Prof. Dr. Felix Wichmann
3. Berichterstatter, falls zutreffend	

To Giorgos

Abstract

Knowledge about causal relationships is important because it enables the prediction of the effects of interventions that perturb the observed system. Specifically, predicting the results of interventions amounts to the ability of answering questions like the following: if one or more variables are *forced* into a particular state, how will the probability distribution of the other variables be affected? Causal relationships can be identified through randomized experiments. However, such experiments may often be unethical, too expensive or even impossible to perform. The development of methods to infer causal relationships from observational rather than experimental data constitutes therefore a fundamental research topic. In this thesis, we address the problem of causal discovery, that is, recovering the underlying causal structure based on the joint probability distribution of the observed random variables.

The causal graph cannot be determined by the observed joint distribution alone; additional *causal* assumptions, that link statistics to causality, are necessary. Under the Markov condition and the faithfulness assumption, conditional-independence-based methods estimate a set of *Markov equivalent* graphs. However, these methods cannot distinguish between two graphs belonging to the same Markov equivalence class. Alternative methods investigate a different set of assumptions. A formal basis underlying these assumptions are functional models which model each variable as a function of its parents and some noise, with the noise variables assumed to be jointly independent. By restricting the function class, e.g., assuming additive noise, Markov equivalent graphs can become distinguishable. Variants of all aforementioned methods allow for the presence of confounders, which are unobserved common causes of two or more observed variables.

In this thesis, we present complementary causal discovery methods employing different kind of assumptions than the ones mentioned above. The first part of this work concerns causal discovery allowing for the presence of confounders. We first propose a method that detects the existence and identifies a finite-range confounder of a set of observed dependent variables. It is based on a kernel method to identify finite mixtures of nonparametric product distributions. Next, a property of a conditional distribution, called purity, is introduced which is used for excluding the presence of a low-range confounder of two observed variables that completely explains their dependence (we call low-range a variable whose range has "small" cardinality).

We further study the problem of causal discovery in the two-variable case, but now assuming no confounders. To this end, we exploit the principle of *independence of causal mechanisms* that has been proposed in the literature. For the case of two variables, it states that, if $X \to Y$ (X causes Y), then $P(X)$ and $P(Y|X)$ do not contain information about each other. Instead, $P(Y)$ and $P(X|Y)$ may contain information about each other. Consequently, estimating $P(Y|X)$ from $P(X)$ should not be possible, while estimating $P(X|Y)$ based on $P(Y)$ may be possible. We employ this asymmetry to propose a causal discovery method which decides upon the causal direction by comparing the accuracy of the estimations of $P(Y|X)$ and $P(X|Y)$.

Moreover, the principle of independence has implications for common machine learning tasks such as semi-supervised learning, which are also discussed in the current work.

Finally, the goal of the last part of this dissertation is to present empirical results on the performance of estimation procedures for causal discovery using Additive Noise Models (ANMs) in the two-variable case.

Experiments on synthetic and real data show that the algorithms proposed in this thesis often outperform state-of-the-art algorithms.

Acknowledgements

First and foremost, I would like to thank my advisors Dominik Janzing and Bernhard Schölkopf for their great mentoring, continuous support and advice. They guided me during my studies in seeking for the right questions to ask but at the same time provided me with the flexibility to define my own research questions.

I would like to thank Dominik for his insightful guidance and constructive feedback. He was always available for my persistent questions which was essential to my progress. I have learned a lot from his perspective on what research is all about. His enthusiasm for pure research was contagious and motivational and he has contributed immensely in making my Ph.D. experience inspirational and challenging.

I am also very grateful to Bernhard for giving me the opportunity to work in such a great and stimulating research environment. I would like to thank him for his inspiration, support and sharp discussions. He is undoubtedly one of the most prominent scientists in the field and I feel very lucky to have collaborated and learned from him. I would further like to sincerely thank the rest of my committee members for their time and help.

Special thanks to Jonas Peters for our collaboration during his time at MPI both as a Ph.D. student and later as a group leader of the causality group. It has always been a pleasure to discuss with him both scientifically and about life perspectives and I truly admire his drive and passion for research. I am also very thankful to my other co-authors Kun Zhang, Philipp Henning, Samory Kpotufe, Joris Mooij and Oliver Stegle and to the rest of my colleagues from MPI for our fruitful scientific discussions. They created a friendly and cooperative working environment and an open atmosphere. Sab-

rina Rehbaum, our secretary, also contributed to this by always being very helpful, efficient and friendly.

I am also very fortunate to have exceptional friends from Athens, my home city. Sofia, Georgia, Manolis, Giannis, Foteini and others, although far, currently scattered around the world, have drastically contributed in making my life balanced and colorful. Moreover, a lot of fun moments of my time in Tübingen are due to my flatmate, Amalia, with whom I shared three fantastic years with endless discussions about life, politics and science.

Very special thanks to my parents and my wonderful sister for their love, constant support and encouragement during my studies. My mother was the first to introduce me to the beautiful world of mathematics and showed me the pleasure of learning and persistence in problem solving. Last but not least, my deep thanks to Giorgos for his long lasting support and patience during these years. His positive attitude, unconditional love and optimism kept me going.

Contents

Chapter 1

Introduction

Machine learning is commonly concerned with prediction tasks [Bishop, 2006, Schölkopf and Smola, 2002, Murphy, 2012], e.g., based on observations of the size and texture of a breast tumor, predict whether it is benign or malignant (binary classification task). However, in many situations, the aim is to uncover the underlying *causal* mechanisms rather than just modeling the observed data. Cause-effect relationships tell us that a specific variable, say whether or not a person smokes, is not just statistically associated with a disease, but it is causal for the disease. Judea Pearl, ACM Turing award recipient in 2011, mentions in his book [Pearl, 2009]: “I now take causal relationships to be the fundamental building blocks both of physical reality and of human understanding of that reality, and I regard probabilistic relationships as but the surface phenomena of the causal machinery that underlies and propels our understanding of the world.” and at a later part “...This puts into question the ruling paradigm of graphical models in statistics according to which conditional independence assumptions are the primary vehicle for expressing substantive knowledge. It seems that if conditional independence judgments are *by-products of stored causal relationships*, then tapping and representing those relationships directly would be a more natural and more reliable way of expressing what we know or believe about the world.”

Besides being a more natural representation, a model built around causal rather than associational information offers the ability to predict the consequences of interventions. An intervention is the action of changing/disturbing

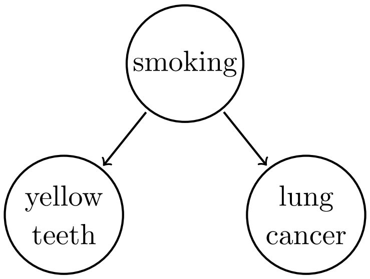


Figure 1.1: An explanation for the correlation between yellow teeth and lung cancer.

the "natural" probability distribution of some of the variables in a system, e.g., setting a given variable to some specified value. Knowledge about causal relationships enables the prediction of the effects of interventions, i.e., prediction of the system reaction in hypothetical experiments that have *not* been performed.

Statistics alone is unable to aid in causal inference: for example, yellow stained teeth may be correlated with lung cancer, however this does not mean that yellow teeth is causal for lung cancer. That is, even though the color of the teeth can be a predictive feature for the presence of lung cancer, nevertheless, if an intervention whitens one's teeth (e.g., by a visit to the dentist), this will not lead to the disappearance of the cancer. Instead, their statistical association could be explained by the presence of a third variable, say smoking, which is a common cause of both. This can be represented by the structure of Fig. 1.1, where arrows indicate causal relations.

One way of obtaining causal knowledge is through randomized trials. In our example, this would correspond to randomly staining the teeth of a part of the population and analyzing the difference in lung cancer between stained and not-stained populations. In the absence of difference, we would conclude that yellow teeth is not causal for lung cancer and seek alternative explanations for their correlation. However, randomized trials often cannot be performed in practice: they may be too expensive, unethical or even impossible. In this case, causal conclusions have to be drawn based solely on observational (and not interventional/experimental) data combined with appropriate *causal* assumptions.

Under the Markov condition and the faithfulness assumption, independence-based methods [Spirtes et al., 2000, Pearl, 2009] estimate a set of directed acyclic graphs (DAGs), all entailing the same set of conditional independences, the so-called *Markov equivalent* graphs. However, these methods cannot distinguish between two graphs belonging to the same Markov equivalence class, e.g. $X \to Y$ and $Y \to X$. Alternative methods investigate a different set of assumptions. A formal basis underlying these assumptions are functional models in which each variable is modeled as a function of its parents and some noise variable. The noise variables are assumed to be jointly independent. Restrictions on the function class, e.g., by assuming additive noise, can lead to distinguishing between graphs belonging to the same Markov equivalence class.

One major challenge of causal discovery is the possible presence of confounders which are unobserved common causes of two or more observed variables. The aforementioned methods, combined with assumptions about the existence of confounders, lead to different results concerning the identifiability of the structure.

This thesis investigates approaches, complementing existing ones, to infer the underlying causal DAG from observational data using various sets of assumptions. Chapter 5 proposes a method to infer the existence and identify a finite-range confounder of a set of observed dependent variables. It is based on a kernel method to identify finite mixtures of nonparametric product distributions. The number of mixture components is found by embedding the joint distribution into a reproducing kernel Hilbert space. The mixture components are then recovered by clustering according to an independence criterion. Chapter 6 is motivated by a problem in genetics. It builds on a property of a conditional distribution $P(Y|X)$, which we call purity. Purity is used as a criterion to infer that the underlying causal structure is $X \to Y$, as opposed to being a DAG containing a low-range latent variable Z in the path between X and Y such that $X \perp\!\!\!\perp Y|Z$ (X independent of Y given Z). Characterizing a conditional as pure is based on the location of the different conditionals $\{P(Y|X = x)\}_x$ in the simplex of probability distributions of Y.

Chapters 7 and 8 use the principle of *independence of causal mechanisms* which has been proposed in the literature. For the case of only two variables, it states that, if $X \to Y$, the marginal distribution of the cause, $P(X)$, and the conditional of the effect given the cause, $P(Y|X)$, are “independent”, in the sense that they do not contain information about each other. Instead, the distribution of the effect, $P(Y)$, and the conditional $P(X|Y)$ may contain information about each other because each of them inherits properties from both $P(X)$ and $P(Y|X)$, hence introducing an asymmetry between cause and effect. This asymmetry has implications for common machine learning tasks such as semi-supervised learning (SSL), discussed in Chapter 7. One more implication of the principle of independence is that estimating $P(Y|X)$ from $P(X)$ should not be possible. However, estimating $P(X|Y)$ based on $P(Y)$ may be possible. Chapter 8 focuses on the problem of causal discovery in the two-variable case, assuming no confounders. Employing the last implication we propose CURE, a causal discovery method which decides upon the causal direction by comparing the accuracy of the estimations of $P(Y|X)$ and $P(X|Y)$ based on the corresponding marginals. To this end, we suggest a method for estimating a conditional based on samples from the marginal, which we call unsupervised inverse GP regression.

Finally, Chapter 9 presents empirical results on the behavior of estimation procedures for causal discovery using additive noise models, also concerning the two-variable case.

1.1 Thesis roadmap

In summary, this dissertation is organized as follows. Chapter 2 provides relevant background and basic concepts necessary throughout the thesis. Chapter 3 introduces the main problems tackled in this dissertation and Chapter 4 is devoted to a literature review of existing causal discovery methods and the assumptions they rely on. Chapters 5 includes a method for identifying a finite-range confounder of a set of observed variables, while Chapter 6 proposes a method for ruling out the existence of a low-range confounder of two observed variables that completely explains their dependence. Chapter 8 is concerned with causal discovery in the two-variable case but now assuming no confounders and is based on the principle of independence of causal mechanisms. This principle has implications also for common machine learning tasks such as SSL, discussed in Chapter 7. Finally, Chapter 9 is concerned with the empirical behavior of estimation methods for ANMs.

This dissertation covers material from the following publications:

- E. Sgouritsa, D. Janzing, J. Peters, and B. Schölkopf. Identifying finite mixtures of nonparametric product distributions and causal inference of confounders. In *Proceedings of the 29th International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013. (Chapter 5)

- D. Janzing, E. Sgouritsa, O. Stegle, J. Peters, and B. Schölkopf. Detecting low-complexity unobserved causes. In *Proceedings of the 27th International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011. (Chapter 6)

- B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012. (Chapter 7)

- E. Sgouritsa, D. Janzing, P. Hennig, and B. Schölkopf. Inference of cause and effect with unsupervised inverse regression. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015. (Chapter 8)

- S. Kpotufe, E. Sgouritsa, D. Janzing, and B. Schölkopf. Consistency of causal inference under the additive noise model. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014. (Chapter 9)

Chapter 2

Background and basic concepts

In this chapter we first provide some background on graphs in Section 2.1 before introducing Bayesian Networks (BNs) (Section 2.2) and causal Bayesian Networks (Section 2.3) to represent probabilistic and causal relationships, respectively. We further discuss an alternative representation using functional models in Section 2.4.

2.1 Graph notation

In the following, we shortly summarize definitions and notations on graphs. Basic graph definitions can be found, for example, in [Spirtes et al., 2000] and [Lauritzen, 1996]. A **graph** G consists of a set of vertices (or nodes) $V = \{1, 2, \dots, d\}$ and a set of edges (or links) $E \subseteq V^2$. If $(a, b) \in E$, then a is said to be a **parent** of b and b a **child** of a, denoted by $a \to b$. The graph $G_1 = (V_1, E_1)$ is called a **proper subgraph** of $G_2 = (V_2, E_2)$ if $V_1 = V_2$ and $E_1 \subset E_2$. The **skeleton** of G is the undirected graph resulting from ignoring all arrowheads in G. Moreover, a **path** is a sequence of distinct vertices $v_1, v_2, \dots, v_n$ such that $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for all $i = 1, \dots, n-1$. A **directed path** is a path $v_1, v_2, \dots, v_n$ such that $(v_i, v_{i+1}) \in E$ for all $i = 1, \dots, n-1$. An **ancestor** of a vertex a is any vertex b such that there is a directed path from b to a. Accordingly, a **descendant** of a is any b such that there is a directed path from a to b. In a path $v_1, \dots, v_n$, v_i is called a

collider if $(v_{i+1}, v_i) \in E$ and $(v_{i-1}, v_i) \in E$. A directed acyclic graph (**DAG**) is a graph in which there is no directed path $v_1, v_2, \ldots, v_n$ with $v_1 = v_n$. A **v-structure** consists of two edges whose arrows point to the same vertex and whose tails are not connected by an edge. A **topological ordering** of a DAG is a sequence $v_1, v_2, \ldots, v_n$ of its vertices such that for every edge $(a, b) \in E$, vertex a comes before vertex b in the ordering.

A path between two vertices a and b is said to be **unblocked** (also called d-connected or open) conditioned on a set of vertices Z, with neither a nor b in Z, if and only if:

1. For every collider w in the path, either w or a descendant of w is in Z
2. No non-collider in the path is in Z

A **blocked** path is a path that is not unblocked.

Definition 1 (d-separation) *Two disjoint sets of vertices A and B are said to be d-separated given a set of vertices Z (also disjoint) if every path between any vertex in A and any vertex in B is blocked conditioned on Z.*

2.2 Bayesian Networks

DAGs have been extensively used to represent a set of random variables and their conditional (in)dependences and came to be known as Bayesian Networks (BNs) [Pearl, 1988]. In a probabilistic graphical model (Bayesian Network), each node $v \in V$ of the graph represents a random variable X_v and the links express probabilistic relations between these variables. We denote random variables with capital letters and their corresponding values with lower case letters, e.g., X and x, respectively. Random vectors are denoted with bold face capital letters and their values with bold face lower case letters, e.g., $\mathbf{X}$ and $\mathbf{x}$, respectively.

Consider d random variables $\mathbf{X} := (X_1, X_2, \ldots, X_d)$[1] with ranges $\mathcal{X}_1, \ldots, \mathcal{X}_d$, respectively, and denote by $P(\mathbf{X})$ their joint distribution. Unless stated

[1] We sometimes overload notation and use $\mathbf{X}$ to also denote the *set* $\{X_1, X_2, \ldots, X_d\}$.

otherwise, assume $P(\mathbf{X})$ has a density $p(\mathbf{x})$ with respect to (w.r.t.) some product measure.

Definition 2 (Markov condition) *The joint distribution $P(\mathbf{X})$ is Markov w.r.t. the DAG G if the following equivalent statements hold:*

- *Markov factorization: $p(\mathbf{x})$ factorizes as follows:*

  $$p(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} p(x_j | \mathbf{pa}_j) \tag{2.1}$$

  where $\mathbf{PA}_j$ is the set of parents of X_j in G.

- *local Markov condition: every variable in G is conditionally independent of its non-descendants given its parents.*

- *global Markov condition:*

  $$\mathbf{A}, \mathbf{B} \text{ } \textit{d-separated given } \mathbf{Z} \textit{ in } G \Rightarrow \mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{Z}$$

  for all $\mathbf{A}, \mathbf{B}, \mathbf{Z}$ disjoint subsets of $\mathbf{X}$.

Definition 3 (Bayesian Network) *A Bayesian Network (BN) over $\mathbf{X}$ is a pair $(G, P(\mathbf{X}))$ such that the joint distribution $P(\mathbf{X})$ is Markov with respect to the DAG G.*

Two DAGs G_1 and G_2 are **Markov equivalent** (or alternatively belong to the same Markov equivalence class) if the set of distributions that are Markov with respect to G_1 coincides with the set of distributions that are Markov w.r.t. G_2. This is the case if the Markov condition entails the same set of conditional independences. Verma and Pearl [1991] show that this happens if and only if the two graphs have the same skeleton and the same set of v-structures. For example, the DAGs $X \to Z \to Y$ and $X \leftarrow Z \leftarrow Y$ are Markov equivalent.

Definition 4 (minimality) *A joint distribution $P(\mathbf{X})$ satisfies minimality with respect to the DAG G if it is Markov w.r.t. G, but not to any proper subgraph of G.*

Definition 5 (faithfulness) *A joint distribution $P(\mathbf{X})$ is faithful with respect to the DAG G if*

$$\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{Z} \Rightarrow \mathbf{A}, \mathbf{B} \text{ \textit{d-separated given} } \mathbf{Z} \text{ \textit{in} } G$$

for all $\mathbf{A}, \mathbf{B}, \mathbf{Z}$ disjoint subsets of $\mathbf{X}$.

In other words, faithfulness assumes that there are no conditional independences that are not entailed by the Markov condition.

2.3 Causal Bayesian Networks

Arrows in causal BNs do not merely represent probabilistic relations, as in BNs, but causal relations. In what follows, we formalize this concept. If an external intervention changes some aspect of the system under consideration, this may lead to a change in the joint distribution $P(\mathbf{X})$. Specifically, an *intervention* corresponds to a real world experiment that changes the “natural” probability distribution of a subset of the variables in $\mathbf{X}$. We denote each such subset by $\mathbf{X}_I := (X_i)_{i \in I}$, with range $\mathcal{X}_I := \times_{i \in I} \mathcal{X}_i$, $I \subseteq \{1, \ldots, d\}$. A special kind of intervention is, for example, the so-called *hard* or *perfect* intervention that forces a variable X to take on a certain value x, symbolized as $do(X = x)$ [Pearl, 2009]. We first focus on the simplest case that $|I| = 1$, i.e., only one variable, say X_i, is intervened on. Then, $P(X_1, X_2, \ldots, X_d | do(X_i = x_i))$, with $i \in \{1, \ldots, d\}$, denotes the joint distribution resulting after intervening on $X_i \in \mathbf{X}$, setting $X_i = x_i$. This is called an *interventional* distribution. The latter is in contrast to the so-called *observational* distribution $P(\mathbf{X})$, which is the joint distribution of $\mathbf{X}$ that we observe before conducting any experiment.

In the more general case, we intervene on more than one variable. Then, $P(X_1, X_2, \ldots, X_d | do(\mathbf{X}_I = \mathbf{x}_I))$ denotes the interventional distribution, resulting after intervening on $\mathbf{X}_I$, setting $\mathbf{X}_I = \mathbf{x}_I := (x_i)_{i \in I}$. Let $\mathbf{P_{do}}(\mathbf{X}) = \{P(X_1, X_2, \ldots, X_d | do(\mathbf{X}_I = \mathbf{x}_I))\}_{(I \in \mathcal{I}) \times (\mathbf{x}_I \in \mathcal{X}_I)}$ denote the set of all possible hard interventional distributions, with $\mathcal{I}$ standing for the power set of $\{1, \ldots, d\}$. Further, let G be a DAG.

Definition 6 (Causal Bayesian Network) *The pair* $(G, P(\mathbf{X}))$ *is called a causal Bayesian Network [Pearl, 2009], if, for every possible interventional distribution* $P(X_1, X_2, \ldots, X_d|do(\mathbf{X}_I = \mathbf{x}_I)) \in \mathbf{P_{do}}(\mathbf{X})$*:*

$$p(x_1, x_2, \ldots, x_d|do(\mathbf{X}_I = \mathbf{x}_I)) = \prod_{\substack{j=1 \\ j \notin I}}^{d} p(x_j|\mathbf{pa}_j) \prod_{i \in I} \delta_{X_i, x_i} \tag{2.2}$$

where $\delta_{X_i,x_i} = \begin{cases} 1 & \textit{if} \quad X_i = x_i \\ 0 & \textit{if} \quad X_i \neq x_i \end{cases}$

The right-hand side of Eq. (2.2) is called a *truncated Markov factorization* [Pearl, 2009], since it is equal to the original Markov factorization (Eq. (2.1)) but with some conditionals "truncated" (removed). According to Eq. (2.2), in a causal Bayesian Network, each interventional distribution (left-hand side) equals a truncated factorization, with the removed conditionals $\{P(X_i|\mathbf{PA}_i)\}_i$ being the ones of the intervened variables $\{X_i\}_{i \in I}$.

It is worth noticing that $\varnothing \in \mathcal{I}$ which corresponds to the special case of no intervention. Hence, the observational distribution $P(\mathbf{X})$ can be considered a special interventional distribution ($P(\mathbf{X}) \in \mathbf{P_{do}}(\mathbf{X})$) with no variable intervened on. In this case, Eq. (2.2) boils down to the (non truncated) Markov factorization of Eq. (2.1).

In the following, we often refer to the DAG G of a causal Bayesian Network $(G, P(\mathbf{X}))$ interchangeably as causal DAG, causal structure or causal graph. Furthermore, the conditional of each variable given its parents, $P(X_j|\mathbf{PA}_j)$, is often referred to as *causal mechanism.* We will henceforth assume that minimality is satisfied (see Definition 4).

The parents $\mathbf{PA}_j$ of a variable X_j in a causal DAG can be thought of as its *direct causes* w.r.t. the set of variables in $\mathbf{X}$. In the special case of a causal DAG with only two variables in which $X \rightarrow Y$, X is called the cause, Y the effect and we simply say that X *causes* Y.

For each intervention $do(\mathbf{X}_I = \mathbf{x}_I)$, the *causal effect* of $\mathbf{X}_I$ on $\mathbf{Y}$, denoted by $p(\mathbf{y}|do(\mathbf{X}_I = \mathbf{x}_I))$, gives the resulting distribution of $\mathbf{Y}$ after the intervention, with $\mathbf{X}_I$ and $\mathbf{Y}$ disjoint subsets of $\mathbf{X}$ [Pearl, 2009, Definition 3.2.1]. Consider two random variables $X, Y \in \mathbf{X}$. If there are x, x' such that $P(Y|do(X = x))$ is different from $P(Y|do(X = x'))$, then we say that X *has a (total) causal effect on* Y.

Interventions are not only limited to hard interventions that set variables to constants. A more general intervention corresponds to changing a causal mechanism $P(X_j|\mathbf{PA}_j)$ to a new one, $\tilde{P}(X_j|\tilde{\mathbf{PA}}_j)$. Then, the truncated factorization in the right-hand side of Eq. (2.2) is replaced by the following factorization:

$$\prod_{\substack{j=1\\ j\notin I}}^{d} p(x_j|\mathbf{pa}_j) \prod_{i\in I} \tilde{p}(x_i|\tilde{\mathbf{pa}}_i)$$

Usually the new set of parents $\tilde{\mathbf{PA}}_j$ is either empty or equals the old one, $\mathbf{PA}_j$. In the former case, the type of intervention is often called *stochastic* [Korb et al., 2004], while in the latter *mechanism change* [Tian and Pearl, 2001] or *parametric* [Eberhardt and Scheines, 2007].

There are several advantages of causal Bayesian Networks over Bayesian Networks. The former are useful for predicting the effects of interventions, without having to *actually* perform the interventional experiment itself. Let $(G, P(\mathbf{X}))$ be a causal Bayesian Network. By Definition 6, each interventional distribution in $\mathbf{P_{do}}(\mathbf{X})$ can be computed just based on the causal DAG G and the observational distribution $P(\mathbf{X})$, without actually performing any experiment. An interventional distribution can be obtained with only minor modifications in the Markov factorization of $P(\mathbf{X})$, specifically, by just replacing the conditionals of the intervened variables. A second advantage of causal BNs is that, roughly speaking, they are more natural and meaningful. For example, a machine learning scientist, not interested in causality, would still consider the graph of Fig. 1.1 a more natural way to encode beliefs about conditional independences than a graph in which the arrow between yellow teeth and smoking is reversed, even though both represent exactly the same conditional independences.

Note that there are cases where there is no causal BN over a set of variables. For example [Peters, 2012, Example 1.6], let $X \leftarrow Z \rightarrow Y$ be the DAG of a causal BN over the variables X, Y, Z. Then, there is no causal BN over only X, Y since there is no DAG satisfying Definition 6 only for these two variables. On the other hand, if $X \rightarrow Z \rightarrow Y$ is the DAG of a causal BN over X, Y, Z, then there is a causal BN over X, Y with DAG $X \rightarrow Y$. In contrast, there are (non causal) BNs over X, Y, with DAGs $X \rightarrow Y$ or $Y \rightarrow X$, in both of the above scenarios, since $P(X, Y)$ is Markov to both of these DAGs.

Proposition 1 (uniqueness) *If there is a causal BN* $(G, P(\mathbf{X}))$ *over* $\mathbf{X}$ *and* $P(\mathbf{X})$ *satisfies minimality w.r.t* G*, then* G *is unique in the sense that there is no other graph* G' *such that* $(G', P(\mathbf{X}))$ *is a causal BN over* $\mathbf{X}$ *[Peters, 2012, Proposition 1.4].*

Definition 7 (causal sufficiency) $X_1, X_2, \ldots, X_d$ *is a causally sufficient set of variables if there is a causal BN over them [Peters, 2012, Definition 1.9].*

An alternative definition of causal sufficiency can be found in the literature,[2] e.g., [Spirtes, 2010]: the random variables in $\mathbf{X}$ are causally sufficient if and only if there is no variable $C \notin \mathbf{X}$ such that C is a direct cause of two or more variables in $\mathbf{X}$ relative to $C \cup \mathbf{X}$. If such a variable C exists, it is called a *confounder*, so causal sufficiency amounts to assuming that there are no confounders.

2.4 Functional models

An alternative way of expressing causal/probabilistic relationships is in the form of functional causal/probabilistic models [Pearl, 2009]. They consist of deterministic functional equations and probabilities are introduced through the assumption that certain variables (noise) in the equations are unobserved.

[2] This definition is slightly different from Definition 7. For an example consult the Appendix.

2.4.1 Functional probabilistic models

A functional probabilistic model (FPM) consists of a set of d equations, one for each $X_j \in \mathbf{X}$:

$$X_j = f_j(\mathbf{PA}_j, N_j), \qquad j = 1, \dots, d, \tag{2.3}$$

where $\mathbf{PA}_j \subseteq \mathbf{X}$, for all j, and $N_1, N_2, \dots, N_d$ represent latent noise variables which are assumed to be jointly independent: $P(\mathbf{N}) = \prod_{j=1}^{d} P(N_j)$.

Drawing directed edges from each variable in $\mathbf{PA}_j$ to X_j, for each j, we obtain a directed graph G corresponding to the FPM. This explains the common symbol $\mathbf{PA}_j$ of this representation with the BN representation of Sections 2.2 and 2.3. In addition, G is required to be acyclic (DAG).

An FPM (for specific functions $f_1, \dots, f_d$, noise distributions $P(N_1), \dots,$ $P(N_d)$, and parents sets $\mathbf{PA}_1, \dots, \mathbf{PA}_d$) induces a unique joint distribution over $\mathbf{X}$: using a topological ordering[3] of the DAG, each variable X_j can be written as a function of the noise variables of the preceding variables. If the induced distribution of an FPM is identical to the joint distribution $P(\mathbf{X})$ that we consider, we say that "the FPM induces/entails a distribution identical to $P(\mathbf{X})$" or, shortly, that "the FPM induces/entails $P(\mathbf{X})$".

Pearl [2009, Theorem 1.4.1] shows that if $P(\mathbf{X})$ is induced by an FPM then it is Markov w.r.t. the DAG G of the FPM. Thus, $(G, P(\mathbf{X}))$ is a Bayesian Network. Moreover, for every Bayesian Network $(G, P(\mathbf{X}))$ there exists an FPM that induces a distribution identical to $P(\mathbf{X})$ (see [Pearl, 2009, p. 31] and references therein). So, we can regard FPMs as an alternative to Bayesian Networks to encode joint distributions: the parent sets define the structure while the functions and noise distributions the parameters (conditional distributions). Note, however, that an FPM contains more information than a BN since many combinations of functions and noise distributions can correspond to the same conditional distributions.

An FPM refers to fixed functions, parent sets and noise distributions of the equations in (2.3), inducing a unique joint distribution. Varying parent

[3] The definition of topological ordering is included in Section 2.1.

sets, functions and/or noise distributions results in different FPMs inducing various joint distributions.

Finally, it should be emphasized that FPMs are purely statistical models, as are Bayesian Networks, and not causal. We describe in the next section functional causal models (a.k.a. structural equation models) which are causal models, as are causal Bayesian Networks. A topological ordering of the DAG corresponding to an FPM does not necessarily correspond to a causal ordering. Instead, the FPM describes $P(\mathbf{X})$ only through the fact that its induced distribution coincides with $P(\mathbf{X})$. An FPM can be alternatively thought of as a set of regression models, one for each variable.

2.4.2 Functional causal models

Functional causal models (FCMs) (often referred to as Structural Equation Models (SEM)) [Pearl, 2009] are the causal counterpart of FPMs, same as causal Bayesian Networks as compared to Bayesian Networks. Specifically, similar to an FPM, a functional causal model consists of a set of d equations, one for each $X_j \in \mathbf{X}$ (Eq. 2.3).

The crucial difference is that a functional causal model $\mathcal{M}$, just like a causal BN, represents the system under interventions [Pearl, 2009]: every interventional distribution $P(X_1, X_2, \ldots, X_d | do(\mathbf{X}_I = \mathbf{x}_I)) \in \mathbf{P_{do}}(\mathbf{X})$ is equal to the distribution induced by the following set of equations:

$$\begin{aligned} X_j = f_j(\mathbf{PA}_j, N_j), &\qquad j \notin I \\ X_i = x_i, &\qquad i \in I \end{aligned}$$

This set of equations is constructed from $\mathcal{M}$ by replacing the equations corresponding to the intervened variables with $X_i = x_i$, $i \in I$, while leaving the rest of equations intact. Thus, if G is the (causal) DAG of an FCM inducing $P(\mathbf{X})$, then $(G, P(\mathbf{X}))$ is a causal Bayesian Network.

We finally note that functional causal models, as defined above, are called *Markovian* causal models by Pearl [2009, p. 30].

2.5 When the graph is known

If the causal graph G is known, for example from prior knowledge, then causal effects (see Section 2.3) can be computed. For example, the causal effect of a variable X on another variable Y is given by the following formula [Pearl, 2009, Theorem 3.2.2] known as *parent adjustment* or *adjustment for direct causes*:[4]

$$p(y|do(X_i = x_i)) = \sum_{\mathbf{pa}_i} p(y|x_i, \mathbf{pa}_i)p(\mathbf{pa}_i)$$

It is enough if the parents of the intervened variable are observed in this case. Yet the more challenging problem is to derive causal effects in situations where some members of $\mathbf{PA}_i$ are unobserved. A causal effect is called identifiable if it can be computed from the observational (pre-intervention) distribution and the graph structure. Graphical tests exist for deciding whether causal effects are identifiable like the back- and front-door criteria [Pearl, 2009]. More generally, the calculus of interventions, the so-called *do-calculus*, was developed by Pearl to facilitate the identification of causal effects. It has been proven to be complete [Shpitser and Pearl, 2006, Huang and Valtorta, 2006], that is, all identifiable causal effects can be computed by an iterative application of its three rules. Moreover, graphical criteria exist [Tian and Shpitser, 2010, Huang and Valtorta, 2006] to find these causal effects.

The literature is rich in what can be achieved in case that the causal graph G is known, but further details fall out of the scope of this thesis. This dissertation concerns, instead, scenarios in which G is unknown and we seek to find it.

[4] The sum could also be an integral.

Chapter 3

Problem statement

The previous chapter motivated the need for causal models: based only on the causal DAG G and the observational distribution $P(\mathbf{X})$, the effects of interventions can be predicted. However, the causal DAG is usually not available and needs to be learned from the observed data, supplemented with additional assumptions. In what follows, we state the general problems concerning (causal) structure learning.

Consider d variables $\mathbf{X} := (X_1, X_2, \dots, X_d)$ and denote by $P(\mathbf{X})$ their joint distribution.

Problem 1 (structure learning) *Consider a Bayesian Network* $(G, P(\mathbf{X}))$ *or a functional probabilistic model with DAG* G *that induces* $P(\mathbf{X})$*. If* G *is unknown, can* G *(or properties of/features of/information about* G*) be recovered from* $P(\mathbf{X})$*? Under what conditions/additional assumptions?*

Clearly, without additional assumptions, G cannot be uniquely recovered from $P(\mathbf{X})$, since there are many DAGs to which $P(\mathbf{X})$ is Markov, e.g., to any fully connected acyclic graph. Imposing appropriate additional assumptions on the set of possible FPMs or BNs can lead to structure-identifiability, explained below:

Definition 8 (structure-identifiability) *A set of BNs is called structure-identifiable if for any two Bayesian Networks* $(G_1, P_1(\mathbf{X}))$ *and* $(G_2, P_2(\mathbf{X}))$

in this set:

$$P_1(\mathbf{X}) = P_2(\mathbf{X}) \Rightarrow G_1 = G_2 \tag{3.1}$$

In other words, structure-identifiability means that the DAG can be uniquely recovered based on the joint distribution.[1] The assumptions made often do not allow one to uniquely determine G but only a *set* of DAGs. We then have identifiability *up to a class* of DAGs. Definition 8 can also be adjusted to refer to FPMs apart from BNs: a set of FPMs is structure-identifiable if for any two FPMs with DAGs G_1 and G_2 inducing distributions $P_1(\mathbf{X})$ and $P_2(\mathbf{X})$, respectively, (3.1) holds.

Problems 2, 3 and 4, that follow, describe variations of Problem 1 when G is a *causal* graph and/or when *latent* variables are allowed. If the DAG G, that we seek for, is a causal DAG then structure learning is referred to as *causal discovery* or *causal structure learning.*

Problem 2 (causal structure learning) *Consider a causal Bayesian Network $(G, P(\mathbf{X}))$ or a functional causal model with graph G that entails $P(\mathbf{X})$. If G is unknown, can G (or properties of G) be recovered from $P(\mathbf{X})$?*[2] *Under what conditions/additional assumptions?*

Let $\mathbf{L} := (L_1, L_2, \ldots, L_l)$ be l unobserved random variables. Denote by $P(\mathbf{X}, \mathbf{L})$ the joint distribution of $(\mathbf{X}, \mathbf{L})$.

Problem 3 (structure learning with latent variables) *Consider a BN $(G, P(\mathbf{X}, \mathbf{L}))$ or a functional probabilistic model with DAG G that induces $P(\mathbf{X}, \mathbf{L})$. If G is unknown, can G (or properties of G) be recovered from $P(\mathbf{X})$? Under what conditions/additional assumptions?*

[1] Structure-identifiability is often referred to, in related literature, simply as identifiability. In this thesis we use this term to discriminate it from parameter-identifiability (see Section 5.2) which means that the model parameters can be uniquely recovered from the joint distribution. Whenever the meaning is clear from the context, we also simply refer to an identifiable model without further specification.

[2] Without conducting any interventional experiments.

Problem 4 (causal structure learning with latent variables) *Consider a causal Bayesian Network* $(G, P(\mathbf{X}, \mathbf{L}))$ *or a functional causal model with graph* G *that induces* $P(\mathbf{X}, \mathbf{L})$*. Can* G *(or properties of* G*) be recovered from* $P(\mathbf{X})$*? Under what conditions/additional assumptions?*

This thesis proposes methods to solve variations of Problems 2 and 4 by considering appropriate additional assumptions. For simplicity, sometimes we first present a method for usual structure learning (Problems 1 or 3) before attaching a causal meaning to it (Problems 2 or 4, respectively). In the former the additional assumptions considered can be viewed as statistical assumptions while in the latter as causal assumptions.

Chapter 4

Literature review

In this chapter, we review various methods for (causal) structure learning. The following approaches tackle the problems of Chapter 3 by considering additional assumptions that render G identifiable (often up to a class of DAGs) from the joint distribution. Section 4.1 deals with Problems 1 and 2, while Section 4.2 concerns Problems 3 and 4. Since there is a lot of related work on structure learning methods, this review is not exhaustive and mainly focuses on methods intended for causal structure learning.

4.1 Structure learning without latent variables

The literature is rich in methods for learning the structure of a Bayesian Network (Problem 1) or a causal Bayesian Network (Problem 2), assuming no latent variables. These can be divided based on the assumptions they make, e.g., faithfulness or additive noise, leading to different structure-identifiability results, e.g., identifiability up or within Markov equivalence classes.

4.1.1 Independence-based methods

To solve Problems 1 and 2, conditional-independence-based methods [Spirtes et al., 2000, Pearl, 2009] (often referred to as constraint-based methods) assume that the observed joint distribution $P(\mathbf{X})$ is not only Markov but also faithful relative to G (see Definition 5). This means that two disjoint subsets of variables $\mathbf{A}$ and $\mathbf{B}$ are conditionally independent given $\mathbf{Z}$ (also disjoint) if and only if $\mathbf{A}$ and $\mathbf{B}$ are d-separated given $\mathbf{Z}$ in G:

$$\mathbf{A}, \mathbf{B} \text{ d-separated given } \mathbf{Z} \Leftrightarrow \mathbf{A} \perp\!\!\!\perp \mathbf{B} | \mathbf{Z}.$$

These methods are based on conditional independences between variables in $\mathbf{X}$: for two variables $X, Y \in \mathbf{X}$, if there exists a subset $\mathbf{Z}$ of $\mathbf{X} \setminus \{X, Y\}$ such that $X \perp\!\!\!\perp Y | \mathbf{Z}$, then there is no edge between X and Y in G. This way the skeleton of G can be found. At a subsequent stage, a number of orientation rules is used to direct *some* of the edges. The output is a graph representing a *set* of Markov equivalent DAGs, all entailing the same set of conditional independences. Graphs within this Markov equivalence class cannot be distinguished without further assumptions. For example, if no conditional independences are observed, in the case of only two variables, $\mathbf{X} = (X, Y)$, constraint-based methods output both $X \to Y$ and $Y \to X$.

Algorithms in this category include the IC [Pearl, 2009], the SGS [Spirtes et al., 2000] and the PC [Spirtes et al., 2000] algorithm. There are differences between them including, but not limited to, the number of required conditional independence tests and the size of the conditioning sets. Conditional independence testing with large conditioning sets is a challenging task in practice.

4.1.2 Bayesian/score-based methods

Score-based methods, e.g., Cooper and Herskovits [1992], Heckerman et al. [1995], Geiger and Heckerman [1994], Heckerman [1995], Chickering [2002], have two basic components: a scoring metric and a search procedure. The metric computes a score for every candidate DAG, reflecting the goodness-of-fit of the structure to the data. In Bayesian methods, the score is proportional to the posterior probability of a structure given the data and any prior knowledge. The search procedure generates networks that are evaluated by the scoring metric. For discrete variables a multinomial likelihood can be used [Cooper and Herskovits, 1992, Heckerman et al., 1995], whereas for continuous variables a linear Gaussian model can be employed [Geiger and Heckerman, 1994]. DAGs that are Markov equivalent receive usually the same score, but there are some exceptions [Cooper and Herskovits, 1992].

Finally, there exist hybrid approaches that combine aspects of both constraint-based and score-based methods, e.g., Tsamardinos et al. [2006], Claassen and Heskes [2012].

4.1.3 Methods restricting the class of functional model

Unless supplemented with domain or expert knowledge, most of the previous structure learning methods cannot, in general, distinguish between DAGs belonging to the same Markov equivalence class (even if few score-based methods assign different scores to DAGs belonging to the same equivalence class, their motivation seems unclear). In order to be able to distinguish between Markov equivalent DAGs (based only on observational data), the approaches presented in this section use the functional model representation (Section 2.4) along with additional appropriate assumptions.

We first focus on Problem 1. Without further assumptions, $P(\mathbf{X})$ could be induced by many DAGs. The idea of this group of methods is to restrict the functions of the FPM. Restricting the function class, restricts the set of distributions that can be induced.

One such restriction is realized using Additive Noise Models (ANMs) proposed by Hoyer et al. [2009] and Peters et al. [2014]. An ANM is an FPM in which the noise is additive, that is the set of equations in (2.3) become:

$$X_j = f_j(\mathbf{PA}_j) + N_j, \qquad j = 1, \ldots, d.$$

Hoyer et al. [2009] and Peters et al. [2014] prove structure-identifiability (see Definition 8) of ANMs, explained below for the simplest case of two variables, $\mathbf{X} = (X, Y)$.

Consider an ANM with DAG $X \to Y$:

$$\begin{aligned} X &= N_X \\ Y &= f(X) + N_Y, \qquad X \perp\!\!\!\perp N_Y \end{aligned}$$

whose induced distribution is $P(X, Y)$. Then, in the generic case (up to some exceptions like the case of linear f and Gaussian X and N_Y), there is no ANM with DAG $Y \to X$ inducing the same joint distribution $P(X, Y)$. That is, there is no function g and noise variable N_X such that $X = g(Y) + N_X$, with $Y \perp\!\!\!\perp N_X$. This means that, in the generic case, the DAG can be uniquely recovered from the joint distribution, i.e., ANMs are structure-identifiable. We often simply say that ANMs are identifiable. The structure learning algorithm then reads: whenever there is an ANM with DAG in one direction (say, $X \to Y$) inducing the joint distribution $P(X, Y)$, but there is no ANM with DAG in the other direction ($Y \to X$) inducing $P(X, Y)$, then the DAG corresponding to the former direction is inferred (in this case $X \to Y$).

The generalization to more than two variables is described in Peters et al. [2014]. Previous work by Shimizu et al. [2006] proves identifiability of ANMs when restricted to linear functions and non-Gaussian input and noise distributions (Linear Non-Gaussian Acyclic Model (LiNGAM)). A generalization of ANMs are the Post-Nonlinear Models (PNL) [Zhang and Hyvärinen, 2009], where $Y = h(f(X) + N_Y)$, with $N_Y \perp\!\!\!\perp X$ and h an invertible function, which are also identifiable, except for some special cases.

The approaches of this category overcome some disadvantages of the previous methods: they allow inference of the DAG within the Markov equivalence class and do not need to assume faithfulness, but only minimality.

Causal counterpart We can use the method above to solve Problem 2 by considering FCMs instead of FPMs. Then, the inferred DAG is the causal DAG G of Problem 2. Janzing and Steudel [2010] justify why causal structure learning using ANMs is reasonable. In particular, they show that if $P(X, Y)$ can be induced by an ANM with DAG $X \to Y$, then the causal DAG $Y \to X$ is unlikely because it would require a specific tuning between the hypothetical distribution of the cause $P(Y)$ and the hypothetical causal mechanism $P(X|Y)$ to generate a distribution that admits an additive noise model from X to Y.[1] Furthermore, Mooij et al. [2014] present empirical

[1] Provided that $P(Y)$ is sufficiently complex.

results providing evidence that additive-noise methods are indeed able to distinguish cause from effect using only purely observational data.

4.1.4 Methods based on the principle of independence of causal mechanisms

To solve Problem 2, other causal inference methods are based on the principle of independence of causal mechanisms [Janzing and Schölkopf, 2010, Lemeire and Dirkx, 2006, Janzing et al., 2012, Daniusis et al., 2010, Schölkopf et al., 2012] which we state below for the simplest case of a causal BN with only two observed variables, assuming no confounders:

Postulate 1 (independence of input and mechanism) *If $X \to Y$, the marginal distribution of the cause, $P(X)$, and the conditional distribution of the effect given the cause, $P(Y|X)$, are "independent" in the sense that $P(Y|X)$ contains no information about $P(X)$ and vice versa.*

The (causal) conditional $P(Y|X)$ can be thought of as the *mechanism* transforming cause X to effect Y. Then, Postulate 1 is plausible if we are dealing with a mechanism of nature that does not care what (input $P(X)$) we feed into it. This independence can be violated in the backward direction: the distribution of the effect $P(Y)$ and the conditional $P(X|Y)$ may contain information about each other because each of them inherits properties from both $P(X)$ and $P(Y|X)$. This constitutes an asymmetry between cause and effect. While Postulate 1 is abstract, the aforementioned approaches provide formalizations by specifying what is meant by *independence* or *information*: Janzing and Schölkopf [2010] postulate *algorithmic* independence of $P(Y|X)$ and $P(X)$, i.e. zero algorithmic mutual information: $I(P(X) : P(Y|X)) \overset{+}{=} 0$. This is equivalent to saying that the shortest description (in the sense of Kolmogorov complexity) of $P(X,Y)$ is given by separate descriptions $P(X)$ and $P(Y|X)$. Since Kolmogorov complexity is uncomputable, practical implementations must rely on other notions of (in)dependence or information.

When causal relations are deterministic, with $Y = f(X)$, $P(Y|X)$ is completely determined by f, so independence between $P(X)$ and $P(Y|X)$ boils down to independence between $P(X)$ and f. For deterministic non-linear relations, Janzing et al. [2012] and Daniusis et al. [2010] define independence through uncorrelatedness between $\log f'$ and the density of $P(X)$ w.r.t. the Lebesgue measure,[2] both viewed as random variables on $[0, 1]$ with uniform measure. This is reformulated in terms of information geometry as a certain orthogonality in information space. The corresponding Information Geometric Causal Inference (IGCI) method sometimes also works for sufficiently small noise. The performance of IGCI on both real-world and simulated data has also been thoroughly studied by Mooij et al. [2014].

Mooij et al. [2010] infer the causal direction by Bayesian model selection, defining non-parametric priors on the distribution of the cause and the conditional of the effect given the cause that favor distributions of low complexity. The motivation of their method stems also from Postulate 1.

4.2 Structure learning with latent variables

This section is mainly concerned with Problem 4: causal discovery with latent variables. Fast Causal Inference (FCI) [Spirtes et al., 2000] extends PC to causal discovery with latent variables. It assumes that the joint distribution $P(\mathbf{X}, \mathbf{L})$ in Problem 4 is, apart from Markov, also faithful relative to G. Based on conditional independences among the observed variables $\mathbf{X}$, it outputs a set of Markov equivalent maximal ancestral graphs (MAGs) [Richardson and Spirtes, 2002]. MAGs are another type of graphs that are closed under marginalization (as opposed to DAGs), a useful property when it comes to latent variables. Claassen et al. [2013] propose FCI+, a more computationally efficient version of FCI.

To distinguish between Markov equivalent graphs, other methods make more assumptions. Silva et al. [2006], apart from faithfulness, make the following assumptions:

- No variable in $\mathbf{X}$ is an ancestor of a variable in $\mathbf{L}$.

[2] Note that a joint density w.r.t. a product measure does not exist in this case.

- The joint distribution of $\mathbf{Y} := (\mathbf{X}, \mathbf{L})$ is induced by a linear ANM:

$$Y_j = \alpha_j \mathbf{PA}_j + N_j, \qquad j = 1, \ldots, d + l.$$

They propose a framework that distinguishes among different causal graphs based on observable tetrad constraints [Silva et al., 2006]. Their contribution is two-fold: their method (1) finds disjoint subsets of the observed variables for which the members of each subset are d-separated by a latent common cause, and (2) finds features of the Markov equivalence class of the latent structure.

Shimizu et al. [2009] extend LiNGAM [Shimizu et al., 2006] for Problems 3 and 4, assuming that $P(\mathbf{X}, \mathbf{L})$ is entailed by a linear ANM with non-Gaussian noise distributions. They further assume that $P(\mathbf{X}, \mathbf{L})$ is faithful to G to output all possible DAGs where each latent variable is a root node and has at least two children.

Finally, Janzing et al. [2009] extend ANMs [Hoyer et al., 2009] for Problems 3 and 4 but for the special case of two observed ($\mathbf{X} = (X, Y)$) and at most one latent variable, i.e., $l = 0$ or $l = 1$, which (if it exists) is a confounder of X and Y. Specifically, their method distinguishes between the following DAGs: $X \to Y$, $Y \to X$ or $X \leftarrow Z \to Y$, with Z an unobserved latent variable (confounder).

Chapter 5

Identifying finite mixtures of nonparametric product distributions and causal inference of confounders

5.1 Introduction

This chapter is concerned with Problems 3 and 4 (structure learning with latent variables). Specifically, the ultimate goal is to detect the existence and identify a finite-range hidden common cause, i.e., confounder, of a set of observed dependent variables. Consider, for example, that we observe three dependent variables X_1, X_2, X_3. The goal is to be able to detect whether or not their dependence is (only) due to a fourth latent variable, in practice of low range[1], say W, that is a common cause of all of them (Fig. 5.1). In case that the DAG of Fig. 5.1 is inferred, we can also recover the full joint distribution $P(X_1, X_2, X_3, W)$, i.e., identify the confounder W.

To this end, we first propose a kernel method to identify finite mixtures of nonparametric product distributions. It is based on a Hilbert space embed-

[1] We call low range a random variable whose range has "small" cardinality.

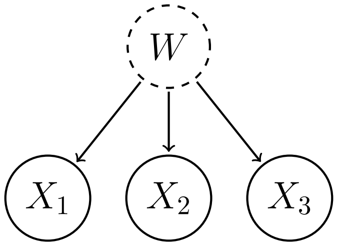


Figure 5.1: Motivating example of DAG to be inferred (the dotted circle represents an unobserved variable).

ding of the observed joint distribution. The rank of the constructed tensor is proven to be equal to the number of mixture components. We present an algorithm to recover the components by partitioning the data points into clusters such that the variables are jointly conditionally independent given the cluster label. We, then, show how this method can be used to identify finite-range confounders.

In Section 5.2, finite mixtures of product distributions are introduced. In Section 5.3, a method is proposed to determine the number of mixture components. Section 5.4 discusses established results on the identifiability of the component distributions. Section 5.5 presents an algorithm for determining the component distributions and Section 5.6 uses the findings of the previous sections for confounder detection and identification. Finally, the experiments are provided in Section 5.7.

5.2 Mixture of product distributions

Consider $d \geq 2$ continuous observed random variables X_1, X_2, …, X_d with ranges $\{\mathcal{X}_j\}_{1\leq j\leq d}$ and assume that their joint distribution $P(X_1, \dots, X_d)$ has a density with respect to the Lebesgue measure. Further, let Z be a finite-range (i.e., that takes on values from a finite set) latent variable[2] with values in $\{z^{(1)}, \dots, z^{(m)}\}$. Only for Sections 5.3-5.5, let $X_1, \dots, X_d$ be jointly conditionally independent given Z, denoted by $X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp \dots \perp\!\!\!\perp X_d \,|\, Z$. This

[2] We often simply say "finite variable" to mean a finite-range variable.

implies the following decomposition of $P(X_1, \dots, X_d)$ into a finite mixture of product distributions:

$$P(X_1, \dots, X_d) = \sum_{i=1}^{m} P(z^{(i)}) \prod_{j=1}^{d} P(X_j | z^{(i)}) \tag{5.1}$$

where $P(z^{(i)}) = P(Z = z^{(i)}) \neq 0$.

By *parameter identifiability* of model (5.1), we refer to the question of when $P(X_1, \dots, X_d)$ uniquely determines the following parameters: (a) the number of mixture components m, and (b) the distribution of each component $P(X_1, \dots, X_d | z^{(i)})$ and the mixing weights $P(z^{(i)})$ up to permutations of z-values.[3] In the next three sections, we focus on determining (a) and (b), when model (5.1) is identifiable. This can be further used to infer the existence of a latent variable confounding a set of observed variables and reconstruct this confounder (Section 5.6).

5.3 Identifying the number of mixture components

Various methods have been proposed in the literature to select the number of mixture components in a mixture model (e.g., Feng and McCulloch [1996], Böhning and Seidel [2003], Rasmussen [2000], Iwata et al. [2013]). However, they impose different kind of assumptions than the conditional independence assumption of model (5.1), e.g., that the distributions of the components belong to a certain parametric family. Assuming model (5.1), Kasahara and Shimotsu [2010] proposed a nonparametric method that requires discretization of the observed variables and provides only a lower bound on m. In the following, we present a method to determine m in (5.1) without making parametric assumptions on the component distributions.

[3] We interchangeably refer to m as the number of mixture components or as the number of states of Z.

5.3.1 Hilbert space embedding of distributions

Our method relies on representing $P(X_1, \ldots, X_d)$ as a vector in a reproducing kernel Hilbert space (RKHS). We briefly introduce this framework. For a random variable X with range $\mathcal{X}$, an RKHS $\mathcal{H}$ on $\mathcal{X}$ with kernel k is a space of functions $f : \mathcal{X} \to \mathbb{R}$ with dot product $\langle \cdot, \cdot \rangle$, satisfying the reproducing property [Schölkopf and Smola, 2002]:

$$\begin{aligned} \langle f(\cdot), k(x, \cdot) \rangle &= f(x), \text{ and consequently,} \\ \langle k(x, \cdot), k(x', \cdot) \rangle &= k(x, x') \end{aligned}$$

The kernel thus defines a map $x \mapsto \phi(x) := k(x, .) \in \mathcal{H}$ satisfying $k(x, x') = \langle \phi(x), \phi(x') \rangle$, i.e., it corresponds to a dot product in $\mathcal{H}$.

Let $\mathcal{P}$ denote the set of probability distributions on $\mathcal{X}$, then we use the following *mean map* [Smola et al., 2007] to define a Hilbert space embedding of $\mathcal{P}$:

$$\mu : \mathcal{P} \to \mathcal{H}; \qquad P(X) \mapsto \mathbb{E}_X[\phi(X)] \tag{5.2}$$

We will henceforth assume this mapping to be injective, which is the case if k is *characteristic* [Fukumizu et al., 2008], as the widely used Gaussian RBF kernel $k(x, x') = \exp(-\|x - x'\|^2 / (2\sigma^2))$.

We use the above framework to define Hilbert space embeddings of distributions of every single X_j. To this end, we define kernels k_j for each X_j, with feature maps $x_j \mapsto \phi_j(x_j) = k(x_j, .) \in \mathcal{H}_j$. We thus obtain an embedding μ_j of the set $\mathcal{P}_j$ into $\mathcal{H}_j$ as in (5.2).

We can apply the same framework to embed the set of joint distributions $\mathcal{P}_{1,\ldots,d}$ on $\mathcal{X}_1 \times \ldots \times \mathcal{X}_d$. We simply define a joint kernel $k_{1,\ldots,d}$ by

$$k_{1,\ldots,d}((x_1, \ldots, x_d), (x'_1, \ldots, x'_d)) = \prod_{j=1}^{d} k_j(x_j, x'_j)$$

leading to a feature map into

$$\mathcal{H}_{1,\ldots,d} := \bigotimes_{j=1}^{d} \mathcal{H}_j$$

with

$$\phi_{1,\ldots,d}(x_1,\ldots,x_d) = \bigotimes_{j=1}^{d} \phi_j(x_j)$$

where $\bigotimes$ stands for the Hilbert space tensor product. We use the following mapping of the joint distribution:

$$\begin{aligned} \mu_{1,\ldots,d} : \mathcal{P}_{1,\ldots,d} &\to \bigotimes_{j=1}^{d} \mathcal{H}_j \\ P(X_1,\ldots,X_d) &\mapsto \mathbb{E}_{X_1,\ldots,X_d}[\bigotimes_{j=1}^{d} \phi_j(X_j)] \end{aligned}$$

5.3.2 Identifying the number of components from the rank of the joint embedding

By linearity of the maps $\mu_{1,\ldots,d}$ and μ_j, the embedding of the joint distribution decomposes into:

$$\mathcal{U}_{X_1,\ldots,X_d} := \mu_{1,\ldots,d}(P(X_1,\ldots,X_d)) = \sum_{i=1}^{m} P(z^{(i)}) \bigotimes_{j=1}^{d} \mathbb{E}_{X_j}[\phi_j(X_j)|z^{(i)}] \quad (5.3)$$

Definition 9 (full rank conditional) *Let A, B be two random variables with ranges $\mathcal{A}, \mathcal{B}$, respectively. The conditional probability distribution $P(A|B)$ is called a full rank conditional if $\{P(A|b)\}_{b\in\mathcal{B}}$ is a linearly independent set of distributions.*

Recalling that the rank of a tensor is the minimum number of rank 1 tensors needed to express it as a linear combination of them, we obtain:

Theorem 1 (number of mixture components) *If $P(X_1,\ldots,X_d)$ is decomposable as in (5.1) and $P(X_j|Z)$ is a full rank conditional for all $1 \leq j \leq d$, then the tensor rank of $\mathcal{U}_{X_1,\ldots,X_d}$ is m.*

Proof. From (5.3), the tensor rank of $\mathcal{U}_{X_1,\ldots,X_d}$ is at most m. If the rank is $m' < m$, there exists another decomposition of $\mathcal{U}_{X_1,\ldots,X_d}$ (apart from (5.3)) as $\sum_{i=1}^{m'} \bigotimes_{j=1}^{d} v_{i,j}$, with non-zero vectors $v_{i,j} \in \mathcal{H}_j$. Then, there exists a vector $w \in \mathcal{H}_1$, s.t. $w \perp \text{span}\{v_{1,1}, \ldots, v_{m',1}\}$ and $w \not\perp \text{span}\{(\mathbb{E}_{X_1}[\phi_1(X_1)|z^{(i)}])_{1\leq i \leq m}\}$. The dual vector $, w\rangle$ defines a linear form $\mathcal{H}_1 \to \mathbb{R}$. By overloading notation, we consider it at the same time as a linear map $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_d \to \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_d$, by extending it with the identity map on $\mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_d$. Then, $\langle \sum_{i=1}^{m'} \bigotimes_{j=1}^{d} v_{i,j}, w\rangle = \sum_{i=1}^{m'} \langle v_{i1}, w\rangle \bigotimes_{j=2}^{d} v_{i,j} = 0$ but $\langle \mathcal{U}_{X_1,\ldots,X_d}, w\rangle \neq 0$. So, $m = m'$. □

The assumption that $P(X_j|Z)$ is a full rank conditional, i.e., $\{P(X_j|z^{(i)})\}_{i\leq m}$ is a linearly independent set, is also used by Allman et al. [2009] (see Section 5.4). It does not prevent $P(X_j|z^{(q)})$ from being itself a mixture distribution, however, it implies that, for all j, q, $P(X_j|z^{(q)})$ is not a linear combination of $\{P(X_j|z^{(r)})\}_{r\neq q}$. Theorem 1 states that, under this assumption, the number of mixture components m of (5.1) (or equivalently the number of values of Z) is identifiable and equal to the rank of $\mathcal{U}_{X_1,\ldots,X_d}$. A straightforward extension of Theorem 1 reads:

Lemma 1 (infinite Z) *If Z takes values from an infinite set, then the tensor rank of $\mathcal{U}_{X_1,\ldots,X_d}$ is infinite.*

Although their connection to causal discovery may not be obvious yet, Theorem 1 and Lemma 1 are used later, in Section 5.6, for detecting the existence of a finite-range confounder.

5.3.3 Empirical estimation of the tensor rank from finite data

Given empirical data for every X_j, $\{x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(n)}\}$, to estimate the rank of $\mathcal{U}_{X_1,\ldots,X_d}$, we replace it with the empirical average

$$\hat{\mathcal{U}}_{X_1,\ldots,X_d} := \frac{1}{n} \sum_{i=1}^{n} \bigotimes_{j=1}^{d} \phi_j(x_j^{(i)}) \,, \tag{5.4}$$

which is known to converge to the expectation in Hilbert space norm [Smola et al., 2007].

The vector $\hat{\mathcal{U}}_{X_1,\ldots,X_d}$ still lives in the infinite dimensional feature space $\mathcal{H}_{1,\ldots,d}$, which is a space of functions $\mathcal{X}_1 \times \cdots \times \mathcal{X}_d \to \mathbb{R}$. To obtain a vector in a finite dimensional space, we evaluate this function at the n^d data points $(x_1^{(q_1)}, \ldots, x_d^{(q_d)})$ with $q_j \in \{1, \ldots, n\}$ (the d-tuple of superscripts $(q_1, \ldots, q_d)$ runs over all elements of $\{1, \ldots, n\}^d$). Due to the reproducing kernel property, this is equivalent to computing the inner product with the images of these points under $\phi_{1,\ldots,d}$:

$$V_{q_1,\ldots,q_d} := \left\langle \hat{\mathcal{U}}_{X_1,\ldots,X_d}, \bigotimes_{j=1}^{d} \phi_j(x_j^{(q_j)}) \right\rangle = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} k_j(x_j^{(i)}, x_j^{(q_j)}) \qquad (5.5)$$

For $d = 2$, V is a matrix, so one can easily find low rank approximations via truncated Singular Value Decomposition (SVD) by dropping low SVs. For $d > 2$, finding a low-rank approximation of a tensor is an ill-posed problem [De Silva and Lim, 2008]. By grouping the variables into two sets, say $X_1, \ldots, X_s$ and $X_{s+1}, \ldots, X_d$ without loss of generality, we can formally obtain the $d = 2$ case with two vector-valued variables. This amounts to reducing V in (5.5) to an $n \times n$ matrix by setting $q_1 = \cdots = q_s$ and $q_{s+1} = \cdots = q_d$. In theory, we expect the rank to be the same for all possible groupings. In practice, we report the rank estimation of the majority of all groupings. The computational complexity of this step is $O(2^{d-1}n^3)$.

5.4 Identifiability of component distributions

Once we have determined the number of mixture components m of model (5.1), we proceed to step (b) (see Section 5.2) of recovering the distribution of each component $P(X_1, \ldots, X_d|z^{(i)})$ and the mixing weights $P(z^{(i)})$. In the following, we describe results from the literature on when these parameters are identifiable, for known m. Hall and Zhou [2003] proved that when $m = 2$, identifiability of parameters always holds in $d \geq 3$ dimensions. For $d = 2$ and $m = 2$ the parameters are generally not identifiable: there is a two-parameter continuum of solutions. Allman et al. [2009] established identifiability of the parameters whenever $d \geq 3$ and for all m under weak conditions[4], using a theorem of Kruskal [1977]. Finally, Kasahara and Shimotsu [2010] provided complementary identifiability results for $d \geq 3$ under different conditions with a constructive proof.

5.5 Identifying component distributions

Theorem 1 states that the number of mixture components m of model (5.1) can be identified with the rank of the Hilbert space embedding of $P(X_1, \ldots, X_d)$. Further, Section 5.4 presented existing results concerning the identifiability of the component distributions $\{P(X_1, \ldots, X_d|z^{(i)})\}_{1 \leq i \leq m}$. In this section, we propose an algorithm that identifies the mixture components. Specifically, consider n data points drawn from $P(X_1, \ldots, X_d)$, with $P(X_1, \ldots, X_d)$ belonging to an identifiable model (5.1). Further, let m be known (it can be estimated as described in Section 5.3.3). Our goal is to cluster the n data points using m labels in such a way that the distribution of points with label i is close to the *unobserved* empirical distribution of every mixture component, $P_n(X_1, \ldots, X_d|z^{(i)})$. In what follows, we often refer to the number of mixture components m as the number of clusters.

5.5.1 Existing methods

Probabilistic mixture models or other clustering methods can be used to identify the mixture components (clusters) (e.g., von Luxburg [2007], Böhning and Seidel [2003], Rasmussen [2000], Iwata et al. [2013]). However, they impose different kind of assumptions than the conditional independence assumption of model (5.1) (e.g., Gaussian mixture model). Assuming model (5.1), Levine et al. [2011] proposed an Expectation-Maximization (EM) algorithm for nonparametric estimation of the parameters in (5.1), given that m is known. Their algorithm uses a kernel as smoothing operator. They choose a common kernel bandwidth for all the components because otherwise their iterative algorithm is not guaranteed not to increase from one iteration to

[4] The same assumption used in Theorem 1, namely that $P(X_j|Z)$ is a full rank conditional for all j.

another. As stated also by Chauveau et al. [2010], the fact that they do not use an adaptive bandwidth [Benaglia et al., 2011] can be problematic especially when the distributions of the components differ significantly.

5.5.2 Proposed method: clustering with independence criterion (CLIC)

The proposed method, CLIC (CLustering with Independence Criterion), assigns each of the n observations to one of the m mixture components (clusters). We do not claim that each single data point is assigned correctly (especially when the clusters are overlapping). Instead, we aim to yield the variables jointly conditionally independent given the cluster (label) in order to recover the mixture components according to model (5.1).

To measure conditional independence of $X_1, \ldots, X_d$ given the cluster we use the Hilbert Schmidt Independence Criterion (HSIC) [Gretton et al., 2008]. It measures the Hilbert space distance between the kernel embeddings of the joint distribution of two (possibly multivariate) random variables and the product of their marginal distributions. If $d > 2$, we test for mutual independence. For that, we perform multiple tests, namely: X_1 against $(X_2, \ldots, X_d)$, then X_2 against $(X_3, \ldots, X_d)$ etc. and use Bonferroni correction. For each cluster, we consider as test statistic the HSIC from the test that leads to the smallest p-value ("highest" dependence).

We regard the negative sum of the logarithms of all p-values (each one corresponding to one cluster) under the null hypothesis of independence as our objective function. The proposed algorithm is iterative. We first randomly assign every data point to one mixture component. In every iteration we perform a greedy search: we randomly divide the data into disjoint sets of c points. Then, we select one of these sets and consider all possible assignments of the set's points to the m clusters (m^c possible assignments). The assignment that optimizes the objective function is accepted and the points of the set are assigned to their new clusters (which may coincide with the old ones). We, eventually, repeat the same procedure for all disjoint sets and this constitutes one iteration of our algorithm. After every iteration we test for conditional independence given the cluster. The algorithm stops after an iteration when any of the following happens: we observe independence in

Algorithm 1 CLIC

```
input data matrix $\mathbf{x}$ of size $n \times d$, $m$, $c$
random assignment $cluster(i) \in \{1, \dots, m\}, i = 1, \dots, n$ of the data into $m$ clusters
while conditional dependence given cluster and clusters change do
    $obj = computeObj(cluster)$
    choose random partition $S_j, j = 1, \dots, J$ of the data into sets of size $c$
    for $j = 1$ to $J$ do
        $newCluster = cluster$
        for all words $w \in \{1, \dots, m\}^c$ do
            $newCluster(S_j) = w$
            $objNew(w) = computeObj(newCluster)$
        end for
        $wOpt = \text{argmin}(objNew)$
        $cluster(S_j) = wOpt$
    end for
end while
if conditional independence given cluster then
    output $cluster$
else
    output “Unable to find appropriate clusters.”
end if
```

all clusters, no data point has changed cluster assignment, an upper limit of iterations has been reached.

The algorithm may not succeed at producing conditionally independent variables for different reasons: e.g., incorrect estimation of m from the previous step or convergence to a local optimum. In that case, CLIC reports that it was unable to find appropriate clusters.

Along the iterations, the kernel test of independence updates the bandwidth according to the data points belonging to the current cluster (in every dimension). Note, however, that this is not the case for the algorithm in Section 5.3. There, we are obliged to use a common bandwidth, because we do not yet have any information about the mixture components.

The parameter c allows for a trade-off between speed and avoiding local optima: for $c = n$, CLIC would find the global optimum after one step, but this would require checking m^n cluster assignments. On the other hand, $c = 1$ leads to a faster algorithm that may get stuck in local optima. In all experiments we used $c = 1$ since we did not encounter serious problems with local optima. Considering c to be a constant, the computational complexity of CLIC is $O(m^c n^3)$ for every iteration. Algorithm 1 includes the pseudocode of CLIC.

5.6 Identifying latent variables/confounders

In this section, we use the results of the previous sections of this chapter for Problems 3 and 4. Before stating our assumptions and main theorem (Theorem 5), we first present some necessary definitions, lemmas and theorems.

Definition 10 (full rank BN) *A BN $(G, P(\mathbf{X}))$ (or an FPM inducing $P(\mathbf{X})$) is called full rank if $P(X_j|\mathbf{PA}_j)$ is a full rank (f.r.) conditional*[5] *for all j.*

The following theorem includes an example of full rank FPM, namely ANM with injective functions:

Theorem 2 (ANM is full rank) *If $P(\mathbf{X})$ is induced by an ANM:*

$$X_j = f_j(\mathbf{PA_j}) + N_j, \qquad j = 1, \ldots, d$$

with $\{N_j\}$ jointly independent and $\{f_j\}$ injective functions, then $\{P(X_j|\mathbf{PA}_j)\}_j$ are full rank conditionals. So, an ANM with injective functions is full rank.

The proof is a straightforward application of Lemma 2:

Lemma 2 (shifted copies) *Let R be a probability distribution on $\mathbb{R}$ and $T_t R$ its copy shifted by $t \in \mathbb{R}$ to the right. Then $\{T_t R\}_{t \in \mathbb{R}}$ are linearly independent.*

[5] See Definition 9.

Proof. Let

$$\sum_{j=1}^{q} \alpha_j T_{t_j} R = 0\,, \tag{5.6}$$

for some q and some q-tuple $\alpha_1, \ldots, \alpha_q$. Let $\hat{R}$ be the Fourier transform of R. If we set $g(\omega) := \sum_{j=1}^{q} \alpha_j e^{i\omega t_j}$ then (5.6) implies $g(\omega)\hat{R}(\omega) = 0$ for all $\omega \in \mathbb{R}$, hence g vanishes for all ω with $\hat{R}(\omega) \neq 0$, which is a set of non-zero measure. Since g is holomorphic, it therefore vanishes for all $\omega \in \mathbb{R}$ and thus all coefficients are zero. □

Lemma 3 (full rank conditional given parent) *If $A \in \mathbf{PA}_X$ is one of the parents of X in a f.r. BN, then, since $P(X|\mathbf{PA}_X)$ is a f.r. conditional (by Definition 10), $P(X|A)$ is also a f.r. conditional (after marginalization).*

Remark: If $A \to B \to C$ is part of the DAG of a f.r. BN, then $P(B|A)$ and $P(C|B)$ are f.r. conditionals (Lemma 3), which implies that $P(C|A)$ is also a f.r. conditional, since it results from their multiplication.

Theorem 3 (rank of parent-child pair) *Assume A is a parent of B in a f.r. BN. Then, the rank of $\mathcal{U}_{A,B}$ is equal to the number of values that A takes, if A is finite. If A is infinite, then the rank of $\mathcal{U}_{A,B}$ is infinite.*

Proof. According to Lemma 3, $P(B|A)$ is a f.r. conditional. Since $A \perp\!\!\!\perp B \,|\, A$ (trivially), applying Theorem 1 for finite $Z := A$ we conclude that the rank of $\mathcal{U}_{A,B}$ is equal to the number of values of A. For infinite A, we similarly apply Lemma 1 and we get infinite rank of $\mathcal{U}_{A,B}$. □

Theorem 4 (rank of d-separated pair) *Assume $A \leftarrow C \to B$ is the DAG of a full rank BN. Then, the rank of $\mathcal{U}_{A,B}$ is equal to the number of values of C.*

Proof. The proof is straightforward: by Definition 10, $P(A|C)$ and $P(B|C)$ are f.r. conditionals. Additionally, $A \perp\!\!\!\perp B \,|\, C$ and then, according to Theorem 1, the rank of $\mathcal{U}_{A,B}$ is equal to the number of values of $Z := C$ (for infinite C, the rank is infinite). □

Theorems 3 and 4 state what is the expected rank of $\mathcal{U}_{A,B}$ for various f.r. BNs. Instead, our goal is to *infer* the structure (see Problems 3 and 4). We first focus on Problem 3. Unlike other methods, we neither make explicit assumptions on the distribution of the variables nor assume faithfulness. Instead, we assume that:

Assumption 1

(a) the Bayesian Network $(G, P(\mathbf{X}, \mathbf{L}))$ (and the FPM) considered in Problem 3 is full rank.

(b) there is at most one (if any) latent variable, i.e., either $l = 1$ or $l = 0$.

(c) latent variables are not descendants of observed ones.

The following theorem uses Theorem 3 to infer G based on the rank of the Hilbert space embedding of the observed joint distribution $P(\mathbf{X})$.

Theorem 5 (identifying latent variables) *Assume that the observed variables $X_1, \ldots, X_d$ are continuous, pairwise dependent. If Assumption 1 holds and the rank of $\mathcal{U}_{X_1,\ldots,X_d}$, with $d \geq 3$, is finite, then Fig. 5.2 depicts the only possible DAG G and $P(X_1, \ldots, X_d, W)$ is identifiable up to reparameterizations of the unobserved variable W.*

Proof. Assume there is at least one edge between two observed variables in G: $X_i \to X_{i'}$. Then, according to Theorem 3, the rank of $\mathcal{U}_{X_i,X_{i'}}$, and thus the rank of $\mathcal{U}_{X_1,\ldots,X_d}$, would be infinite. Therefore, edges between the $\{X_j\}$ can be excluded. Then, the statistical dependences between the $\{X_j\}$ can only be explained by latent variables. Since G, by Assumption 1(b), has at most one latent variable and the observed variables are pairwise dependent, the only possible f.r. DAG is depicted in Fig. 5.2 (with $\mathbf{L} = W$). This implies $X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp \ldots \perp\!\!\!\perp X_d \,|\, W$ (according to the Markov condition), so model (5.1) holds, with $Z := W$ being the latent variable. According to the previous sections (Theorem 1 and Section 5.4), this model is identifiable. □

Based on Theorem 1, the number of values of W is equal to the rank of $\mathcal{U}_{X_1,\ldots,X_d}$ and $P(X_1, \ldots, X_d, W)$ can be identified according to Section 5.5.

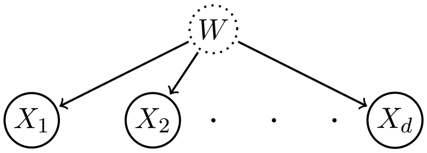


Figure 5.2: Inferred DAG G (the dotted circle represents an unobserved variable).

Note that the single latent variable W could be the result of merging many latent variables $W_1, \ldots, W_k$ to one vector-valued variable $\mathbf{W}$. Thus, at first glance, it seems that one does not lose generality by assuming only one latent. However, Assumption 1(a), then, excludes the case where $\mathbf{W}$ consists of components each of which only acts on some different subset of the $\{X_j\}$. $W_1, \ldots, W_k$ should all be parents of all $\{X_j\}$.

Note that Theorem 5 solves Problem 3 under Assumption 1, when the rank of $\mathcal{U}_{X_1,\ldots,X_d}$ is finite. In contrast, if the rank is infinite, no structure is inferred: infinite rank can be due to edges between the observed variables and/or due to continuous latent variables, etc.

Since we are given only finite data, the estimated rank of $\mathcal{U}_{X_1,\ldots,X_d}$ is always finite, highly depending on the strength of the dependences and the sample size. Then, we are faced with the issue that, based on Theorem 5, we would always infer that Fig. 5.2 depicts the only possible f.r. BN, with the number of values of W being equal to the estimated rank. However, the lower the rank, the more confident we get that this is, indeed, due to the existence of a latent variable that renders the observed variables conditionally independent (Fig. 5.2). On the other hand, high rank can also be due to edges between the observed variables or continuous latent variables. For that, we consider Theorem 5 to be more appropriate for inferring the existence of a latent variable with a small number of values which would lead to low rank. However, we admit that what is considered "high" or "low" is not well defined. For example, how much "high" rank values we expect for the DAG $X_1 \to X_2$ depends on the strength of the dependence: roughly speaking, low dependence between X_1 and X_2 could lead to low estimated rank. In practice, we could make a vague suggestion that whenever the estimated rank is below 5 (although the dependence between the $\{X_j\}$ is strong), it is quite possible that this is due to a latent variable (Fig. 5.2) but for higher rank it is getting more difficult to decide upon the underlying structure.

Causal counterpart Using Assumption 1 for the causal BN considered in Problem 4, Theorem 5 gets directly applicable to Problem 4. In this case, the inferred DAG of Fig. 5.2 is the causal DAG G of Problem 4 and the latent variable W is a confounder.

5.7 Experiments

We conduct experiments both on simulated and real data. In all our experiments we use a Gaussian RBF kernel $k(x, x') = \exp\left(-\|x - x'\|^2/(2\sigma^2)\right)$. Concerning the first step of determining the number of mixture components: a common way to select the bandwidth σ_j for every k_j is to set it to the median distance between all data points in the jth dimension of the empirical data. However, this approach would usually result in an overestimation of the bandwidth, especially in case of many mixture components (see also [Benaglia et al., 2011]). To partially account for this, we compute the bandwidth for every X_j as the median distance between points in the neighborhood of every point in the sample. The neighborhood is found by the 10 nearest neighbors of each point computed using all other variables apart from X_j. To estimate the rank of V, we find its SVD and report the estimated rank as $\hat{m} = \text{argmin}_i(SV_{i+1}/SV_i)$ within the SVs that cover 90-99.999% of the total variance. Finally, concerning CLIC, we use 7 as the maximum number of iterations, but usually the algorithm terminates earlier.

5.7.1 Simulated data

Simulated data are generated according to the DAG of Fig. 5.2 (we henceforth refer to them as the first set of simulated data), i.e., model (5.1) holds with $Z := W$, since $X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp \ldots \perp\!\!\!\perp X_d \,|\, W$. We first generate Z from a uniform distribution on m values. Then, the distribution $P(X_j|z^{(i)})$ of each mixture component in every dimension is chosen randomly between: (i) a normal distribution with standard deviation 0.7, 1, or 1.3, (ii) a t-distribution with degrees of freedom 3 or 10, (iii) a (stretched) beta distribution with alpha

0.5 or 1 and beta 0.5 or 1, and (iv) a mixture of two normal distributions with variance 0.7 for each. The distance between the components in each dimension is distributed according to a Gaussian with mean 2 and standard deviation 0.3. We choose the distance and the mixtures such that the experiments cover different levels of overlap between the components and at the same time $\{P(X_j|z^{(i)})\}_{i\leq m}$ are generically linearly independent. This way the assumptions of Theorem 1 are satisfied so we expect the rank of $\mathcal{U}_{X_1,\ldots,X_d}$ to be m. We run 100 experiments for each combination of $d = 2, 3, 5$ and $m = 2, 3, 4, 5$, with the sample size being $300 \times m$.

For comparison, we additionally generate data where there are edges also *between* the observed variables and thus are conditionally dependent given the confounder (we henceforth refer to them as the second set of simulated data). For that, we first generate data according to the DAG of Fig. 5.2, as above, for $d = 2$ and $m = 1$ (which amounts to no confounder) and for $d = 2$ and $m = 3$ (3-state confounder). X_2 is then shifted by $4X_1$ to simulate $X_1 \to X_2$. In this case, according to Theorem 3, the rank of $\mathcal{U}_{X_1,X_2}$ is infinite.

Identifying the number of mixture components

We first report results on the first part of identification, i.e. identifying the number of mixture components m of (5.1). The empirical rank estimation may depend on the strength of the dependences, the kernel bandwidth selection, the sample size and the way to estimate the rank by keeping only large eigenvalues. Figures 5.3, 5.4, 5.5 and 5.6 illustrate histograms of the esti-

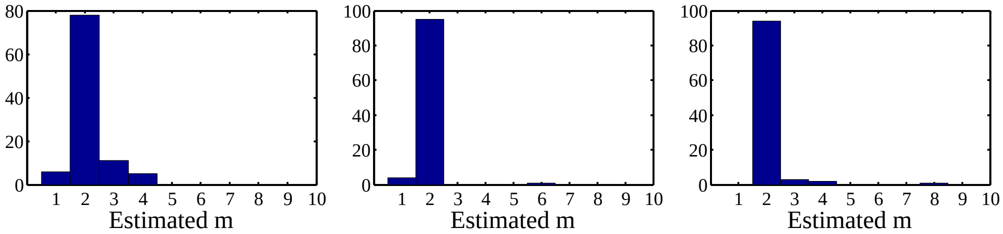


Figure 5.3: Histograms of the estimated number of mixture components m for the first set of simulated data, for $m = 2$ throughout, and $d = 2$ (left), $d = 3$ (middle), $d = 5$ (right).

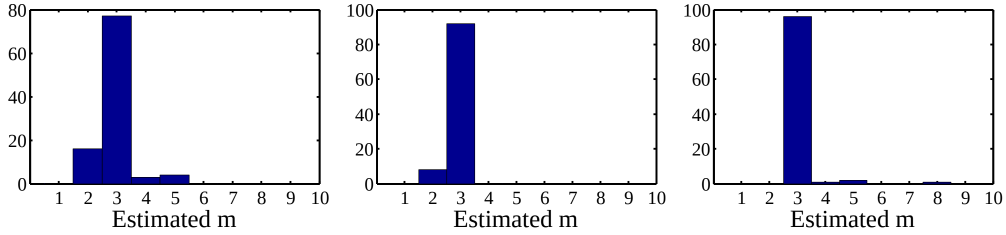


Figure 5.4: As Fig. 5.3 but for $m = 3$.

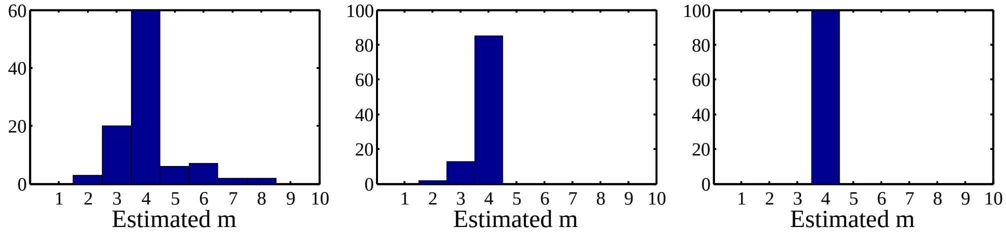


Figure 5.5: As Fig. 5.3 but for $m = 4$.

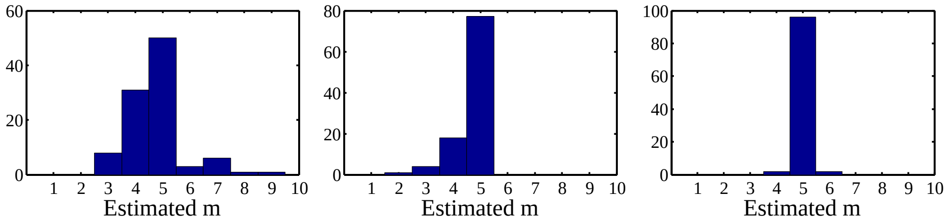


Figure 5.6: As Fig. 5.3 but for $m = 5$.

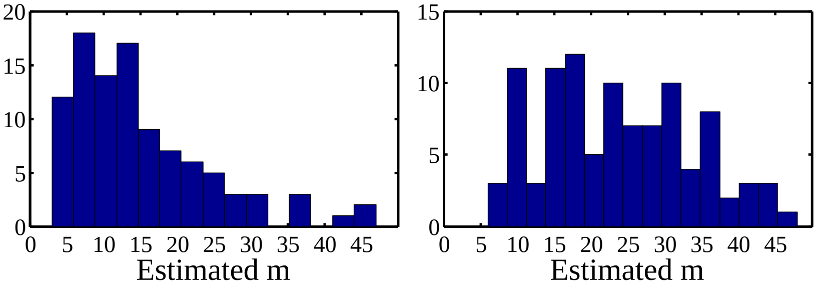
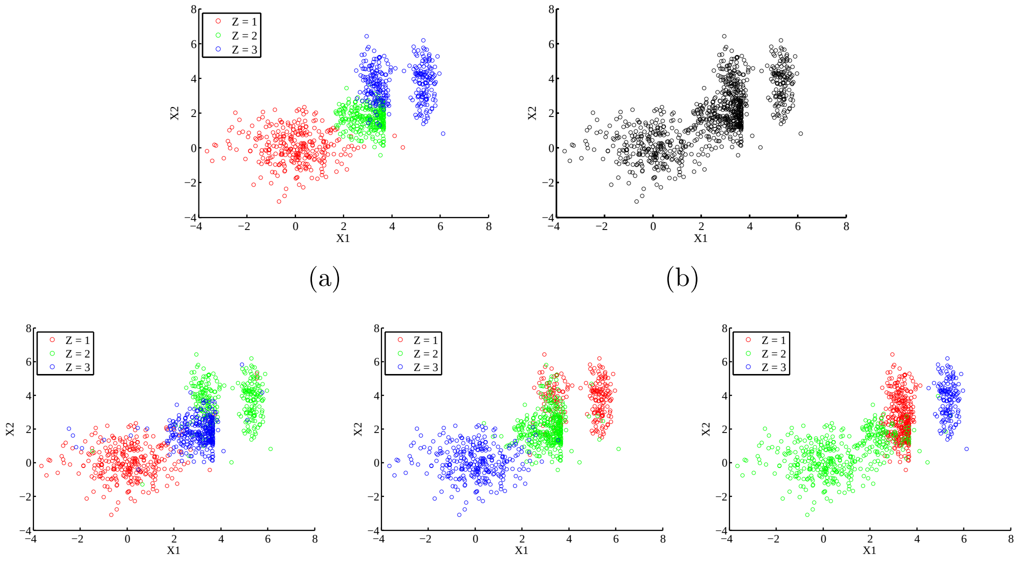


Figure 5.7: Histograms of the estimated m (estimated rank of $\mathcal{U}_{X_1,X_2}$) for the second set of simulated data, i.e., with an edge between the observed variables. Left: no confounder, right: 3-state confounder. As expected, we get relatively "high" values as compared to the estimated m for the first set of simulated data (see Figs. 5.3-5.6).

mated number of mixture components (equivalently the estimated number of values of the confounder) for the first set of simulated data for $m = 2, 3, 4$ and 5, respectively. Each figure contains one histogram for every value of $d = 2, 3$ and 5. We can observe that as m increases the method becomes more sensitive in underestimating the number of components, a behavior which can be explained by the common sigma selection for all the data in each dimension or by high overlap of the distributions (which could violate Assumption 1(a)). On the other hand, as d increases the method becomes more robust in estimating m correctly due to the grouping of variables that allows multiple rank estimations. The "low" estimated rank values provide us with some evidence that the DAG of Fig. 5.2 is true (Theorem 5). Of course, as stated also at the end of Section 5.6, it is difficult to define what is considered a low rank.

Figure 5.7 depicts histograms of the estimated number of mixture components for the second set of simulated data. According to Theorem 3, the edge $X_1 \to X_2$ results in an infinite rank of $\mathcal{U}_{X_1,X_2}$. Indeed, we can observe that in this case the estimated m is much higher. The "high" estimated rank values provide us with some evidence that the underlying DAG may include edges between the observed variables or confounders with a high or infinite number of values. Note that, depending on the strength of the dependence between X_1 and X_2, we may get higher or lower rank values. For example,



(a)

(b)

(c)

(d)

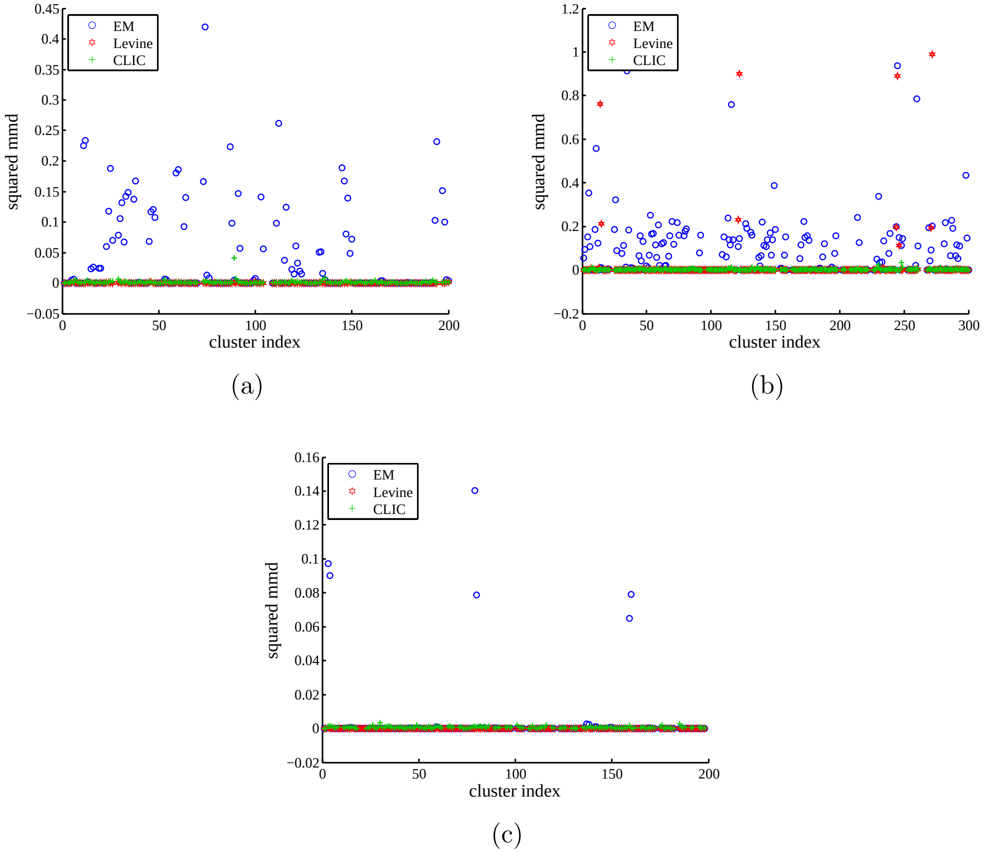
(e)

Figure 5.8: (a) Ground truth, (b) input, (c) CLIC output, (d) Levine output, and (e) EM output for simulated data generated for $m = 3$ and $d = 2$. Each color represents one mixture component. EM incorrectly merges two clusters since it assumes a Gaussian mixture model and not model (5.1), as opposed to CLIC and Levine methods.

if the strength is very weak we get lower rank values since the dependence between X_1 and X_2 tends to be dominated by the confounder (that has a small number of values).

Full identification framework

Next, we perform experiments using the first set of simulated data to evaluate the performance of the proposed clustering method (CLIC) (Section 5.5.2), the method of Levine et al. [2011] (Section 5.5.1) and the EM algorithm using a Gaussian mixture model (EM is repeated 5 times and the solution with the largest likelihood is reported). In the following, we refer to these methods as CLIC, Levine, and EM, respectively. For each data point, the



(a)

(b)

(c)

Figure 5.9: Squared MMD between output and ground truth clusters, for each of the three methods, for simulated data with (a) $d = 3, m = 2$, (b) $d = 3, m = 3$ and (c) $d = 5, m = 2$. CLIC and Levine methods perform significantly better than EM, since they assume model (5.1).

two latter methods output posterior probabilities for the m clusters, which we sample from to obtain cluster assignments. Figure 5.8 illustrates the cluster assignments of these three methods for one simulated dataset[6] with $m = 3$ and $d = 2$. Note that permutations of the colors are, as expected, due to the ambiguity of labels in the identification problem. However, EM incorrectly

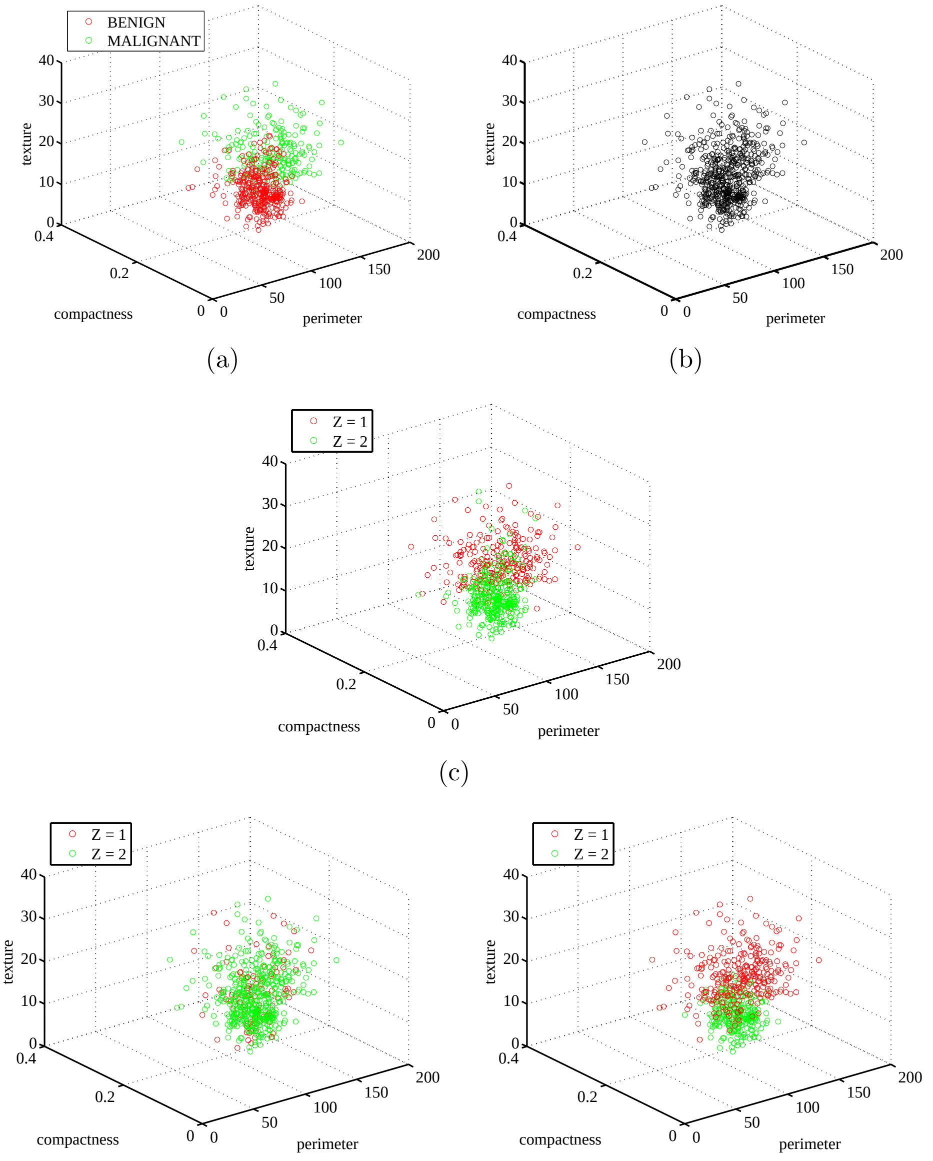
[6] This example is intended for visualization purposes only, because for these values of d and m model (5.1) is not always identifiable according to Section 5.4.

identifies a single component (having a mixture of two Gaussians as marginal density in X_1 dimension) as two distinct components. It is clear that this is because it assumes that the data are generated by a Gaussian mixture model and not by model (5.1), as opposed to CLIC and Levine methods.

We compare the distribution of each cluster output, for each of the three methods, to the empirical distribution, $P_n(X_1, \ldots, X_d|z^{(i)})$, of the corresponding mixture component (ground truth). For that we use the squared maximum mean discrepancy (MMD) [Gretton et al., 2012] that is the distance between Hilbert space embeddings of distributions. We only use the MMD and not one of the test statistics described in [Gretton et al., 2012], since they are designed to compare two independent samples, whereas our samples (output and ground truth) have overlapping observations. To account for the permutations of z-values, we measure the MMD for all cluster permutations and select the one with the minimum sum of MMD for all clusters. Figures 5.9(a)-5.9(c) report the squared MMD results of the three methods for different combinations of m and d. Each point corresponds to the squared MMD for one cluster of one of the 100 experiments. Results are provided only for the cases that the number of components m is correctly identified from the previous step. The CLIC method is unable to find appropriate clusters in 2 experiments for $d = 3$ and $m = 3$ and in 13 for $d = 5$ and $m = 2$. Without claiming that the comparison is exhaustive, we can infer that both CLIC and Levine methods perform significantly better than EM, since they impose conditional independence. For higher d, EM improves since the clusters are less overlapping. However, the computational time of CLIC is higher compared to the other two methods.

5.7.2 Real data

We further apply our framework to the Breast Cancer Wisconsin (Diagnostic) dataset from the UCI Machine Learning Repository [Bache and Lichman, 2013]. The dataset consists of 32 features of breast masses along with their classification as benign (B) or malignant (M). The sample size of the dataset is 569 (357 B, 212 M). We select 3 features, namely perimeter, compactness and texture, which are pairwise dependent (the minimum p-value is $pval = 2.43e - 17$), but become (close to) mutually independent when we condition on the class (B or M) ($pval_B = 0.016, pval_M = 0.013$). We apply our method

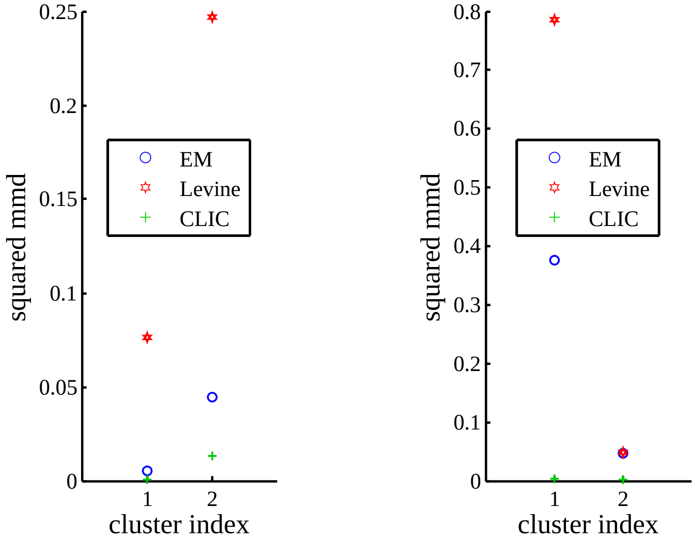


(a) (b)

(c)

(d) (e)

Figure 5.10: (a) Ground truth, (b) input, (c) output CLIC, (d) Levine, and (e) EM for the breast data. Levine's method clustering is far from the ground truth for this dataset.



(a) (b)

Figure 5.11: Squared MMD between output and ground truth clusters for (a) breast and (b) arrhythmia data.

to these three features (assuming the class is unknown) and we succeed at correctly inferring that the number of mixture components is 2. Figure 5.10 depicts the ground truth of the breast data, the input and the results of CLIC, Levine and EM, and Fig. 5.11(a) the corresponding squared MMDs. We can observe that Levine method performs very poorly for this dataset.

Additionally, we select different features, namely perimeter and area, and concavity and area, which are not conditionally independent given the binary class. In this case, we get rank values higher than two, in particular 62 and 8, respectively (Fig. 5.12).

We similarly apply our framework to the Arrhythmia dataset (sample size 452)[Bache and Lichman, 2013]. We select 3 features, namely height, QRS duration and QRSTA of channel V1 which are dependent (minimum $pval = 8.96e-05$), but become independent when we condition on a fourth feature, the sex of a person (male or female) ($pval_M = 0.0607, pval_F = 0.0373$). We

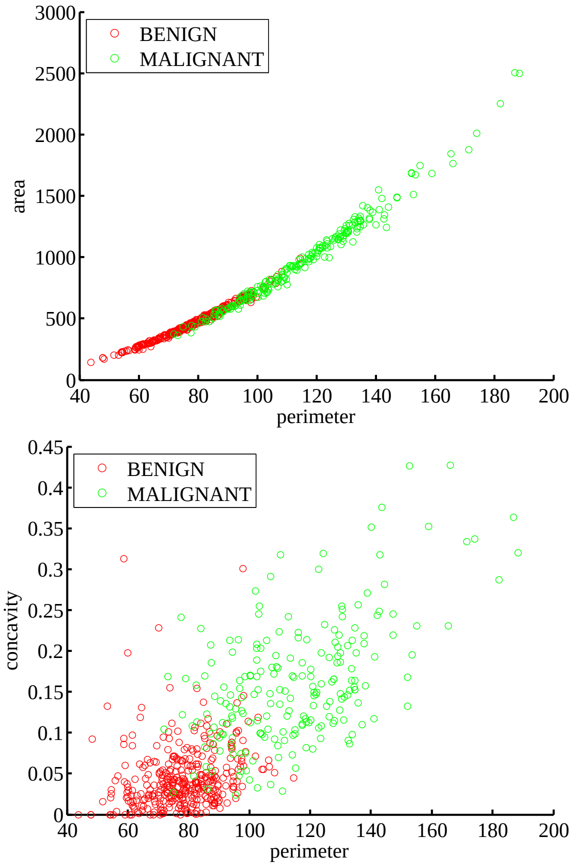
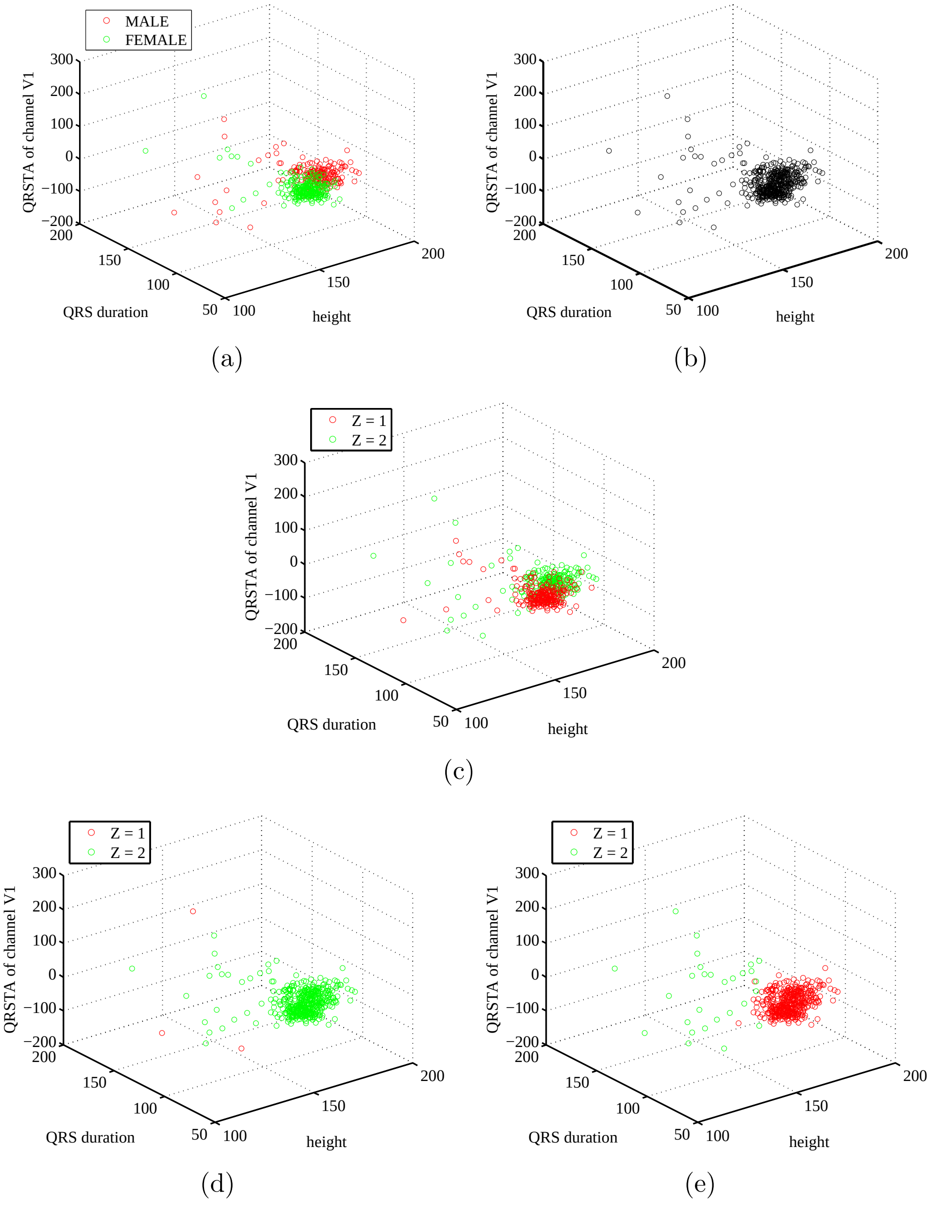


Figure 5.12: Breast data: features conditionally dependent given the class. In this case the estimated m is much higher than 2. Top: estimated $m = 62$, bottom: estimated $m = 8$.

apply our method to the three features and succeed at correctly inferring that the number of mixture components is 2. Figure 5.13 depicts the ground truth, the input and the results of CLIC, Levine and EM, and Fig. 5.11(b) the corresponding squared MMDs. We can observe that Levine and EM methods perform very poorly for this dataset.

Finally, we apply our method to a database with cause-effect pairs[7] (version 0.8), a detailed description of which has recently been provided by Mooij

[7] http://webdav.tuebingen.mpg.de/cause-effect/



(a)

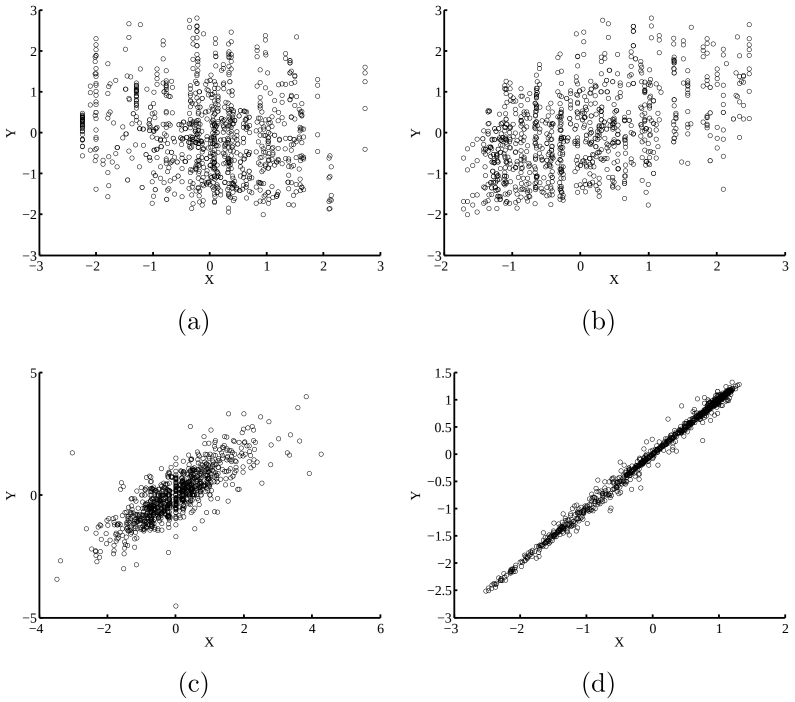
(b)

(c)

(d)

(e)

Figure 5.13: (a) Ground truth, (b) input, (c) output CLIC, (d) Levine, and (e) EM for the arrhythmia data. Both Levine and EM methods perform very poorly for this dataset.



(a) (b) (c) (d)

Figure 5.14: Four cause-effect pairs. Estimated m: (a) $m = 1$, (b) $m = 4$, (c) $m = 8$, and (d) $m = 63$.

et al. [2014]. It includes pairs of variables from various domains with known causal structure, $X \rightarrow Y$. Since $X \rightarrow Y$, we expect the rank of $\mathcal{U}_{X,Y}$ to be infinite given our assumptions (Theorem 3), even if there exist hidden confounders. However, the estimated rank from finite data is always finite, its magnitude strongly depending on the strength of the dependence and the sample size, as mentioned in Section 5.6. Figure 5.14 depicts 4 cause-effect pairs with the same sample size (1000 data points) but various degrees of dependence, specifically: (a) $pval = 7.16e - 12$, (b) $pval = 9.41e - 63$, (c) $pval = 1.21e - 317$ and (d) $pval \approx 0$. The estimated ranks are $m = 1, 4, 8$ and 63, respectively. Note that when X and Y are close to independent (e.g., Fig. 5.14(a)) the assumption of pairwise dependence of Theorem 5 is almost violated.

5.8 Conclusion

In this chapter, we introduce a kernel method to identify finite mixtures of nonparametric product distributions. The method is further used to infer the existence and identify a finite hidden common cause of a set of observed variables. Experiments on simulated and real data were performed for evaluation of the proposed approach. The proposed method has the advantage of being nonparametric. On the downside, it is difficult to arrive at definite conclusions from finite data and the method is in practice more appropriate for the identification of a confounder with a small number of states.

Chapter 6

Ruling out the existence of confounders

6.1 Introduction

The findings of this chapter are complementary to those of the previous, again concerning a specific variant of Problems 3 and 4. Particularly, our goal is, based on $P(X, Y)$, to distinguish between $X \to Y$ and DAGs in which there exists a low range hidden variable Z in the path between X and Y that d-separates them, e.g. $X \leftarrow Z \to Y$.

The motivation stems from statistical genetics. An important problem in biology and medicine is to find genetic causes of phenotypic differences among individuals. Let Y describe a phenotypic difference among individuals such as the presence or absence of a disease, the size of a plant, or the expression level of a gene. These phenotypes are known to correlate with polymorphic loci in the genome, such as single-nucleotide polymorphisms (SNPs). However, due to the strong dependences among nearby SNPs, it is hard to identify those that influence the phenotype. Given a SNP X that is correlated with a phenotype of interest Y, we want to detect whether this marker is causal or it only correlates with a causal one. Specifically, the task is to decide whether the dependence between X and Y is because X influences Y or only due to statistical dependence between X and some other unobserved SNPs Z

influencing Y. Z could be also some environmental condition that influenced X (via evolution) and Y. Thus, we have either

1. $X \to Y$ or
2. $X \leftarrow Z \to Y$ or $X \leftrightarrow Z \to Y$, where $\leftrightarrow$ symbolizes that X and Z are related by a common cause.

We assume that Z in the second category is a low range variable, which is the case if, for example, it describes only a small number of SNPs, each of which is a binary variable. Note that domain knowledge excludes $Y \to X$. Moreover, cases in which both categories 1. and 2. hold, e.g., $X \to Y$ *and* they are confounded by Z, are also included in the first category, since the goal is to decide whether SNP X influences Y. In case of the first category, we call X a causal SNP and (X, Y) a causal pair while in the second category X is called a non-causal SNP and (X, Y) a non-causal pair.[1]

We propose a method to distinguish the first DAG ($X \to Y$) from the rest where a low range unobserved variable Z d-separates X and Y. The proposed method is based on a property of conditionals which we call *purity*. The characterization of a conditional as pure depends on the location of the conditional distributions $\{P(Y|X = x)\}_x$ in the simplex of all probability distributions of Y. A pure conditional $P(Y|X)$ excludes the existence of a low range unobserved variable that d-separates X and Y, thus leading to $X \to Y$.

Purity is introduced in Section 6.2 and Section 6.3 describes how to estimate it from finite data. Section 6.4 includes experimental results followed by a conclusion in Section 6.5.

6.2 Pure conditionals

We introduce a property of a conditional distribution $P(Y|X)$, called purity. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ denote the ranges of three random variables X, Y, Z, respectively,

[1] This naming is due to the role of SNP X with respect to the phenotype Y: in the first category X is causal for Y, while in the second it is not.

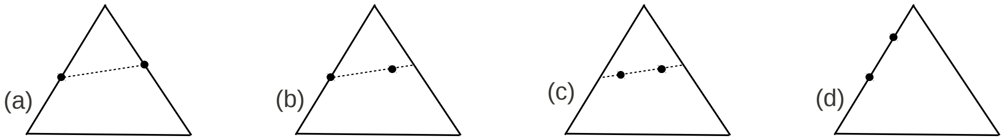


Figure 6.1: Visualization of the location of different $\{P(Y|X=x)\}_x$ in the simplex $\mathcal{P}_Y$, here for $|\mathcal{Y}|=3$: (a) pairwise pure, because the line connecting $P(Y|X=x_1)$ and $P(Y|X=x_2)$ (the black dots) cannot be extended without leaving the simplex; (b) one-sided pairwise pure; (c) and (d) are not pairwise pure, although both points in (d) are not in the interior of $\mathcal{P}_Y$.

and $\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z$ denote the simplex of probability distributions on these sets, respectively. Clearly, $P(Y|X=x) \in \mathcal{P}_Y$ for every $x \in \mathcal{X}$ and also every *convex* combination of distributions $\{P(Y|X=x)\}_x$ lies in $\mathcal{P}_Y$. Whether also *affine* combinations that contain some negative coefficients yield distributions in $\mathcal{P}_Y$ is an interesting property of $P(Y|X)$. We assume that $P(X,Y)$ has a density $p(x,y)$ w.r.t. a product measure.

Definition 11 (pure conditional)
A conditional $P(Y|X)$ is called k-wise pure if for every k-tuple of different x-values $(x_1,\dots,x_k)$ the following condition holds: for all $\lambda \in \mathbb{R}^k \setminus [0,1]^k$ with $\sum_j \lambda_j = 1$

$$\exists y : \quad \sum_{j=1}^{k} p(y|x_j)\lambda_j < 0$$

We also say "pairwise pure" instead of "2-wise pure". $P(Y|X)$ is called one-sided pairwise pure if for every pair (x_1,x_2) with $x_1 \neq x_2$ and for all $\mu < 0$ either

$$\begin{aligned} \exists y : &\quad \mu p(y|x_1) + (1-\mu)p(y|x_2) < 0 \\ \textit{or } \exists y : &\quad \mu p(y|x_2) + (1-\mu)p(y|x_1) < 0 \end{aligned}$$

(see Fig. 6.1 for some examples).

Lemma 4 (quotient of densities)
$P(Y|X)$ is pairwise pure if and only if for every pair (x_1, x_2) with $x_1 \neq x_2$

$$\inf_{y \in \mathcal{Y}\,:\, p(y|x_2) \neq 0} \frac{p(y|x_1)}{p(y|x_2)} = 0\,. \tag{6.1}$$

One-sided pairwise purity holds if and only if, for every pair (x, x'), (6.1) holds either for $x_1 = x$ and $x_2 = x'$ or for $x_1 = x'$ and $x_2 = x$.

Proof. If (6.1) does not hold we set $c := \inf_y p(y|x_1)/p(y|x_2)$ with $0 < c < 1$. Then choosing the coefficient $\mu = 1/(1-c)$ (such that $1 - \mu$ is negative) ensures

$$\mu p(y|x_1) + (1-\mu) p(y|x_2) \geq 0\,, \tag{6.2}$$

for all y with $p(y|x_2) \neq 0$. If $p(y|x_2) = 0$, the left hand side of (6.2) is non-negative anyway. Hence, purity is violated. On the other hand, if $P(Y|X)$ is not pure there is by definition a pair (x_1, x_2) and $\mu < 0$ such that $(1-\mu)p(y|x_1) + \mu p(y|x_2) \geq 0$ for all y, then $\frac{p(y|x_1)}{p(y|x_2)} \geq \frac{-\mu}{1-\mu}$, which contradicts (6.1). □

The following theorem states that if a conditional $P(Y|X)$ is pairwise pure then the existence of an unobserved variable (with compact range and $P(Z|x)$ having continuous strictly positive densities) that d-separates X and Y can be excluded, thus leading to $X \to Y$.

Theorem 6 (excluding compact Z)
If $P(Y|X)$ is pairwise pure then there is no variable Z with compact range and all $\{P(Z|x)\}_x$ having continuous strictly positive densities such that $X \perp\!\!\!\perp Y|Z$.

Proof. Assume such a variable Z existed. Since $\{P(Z|x)\}_x$ have continuous strictly positive densities and Z has compact range, the conditional $P(Z|X)$ is not pairwise pure according to Lemma 4 (see also Lemma 3 of Janzing et al. [2011]). This means that, according to Definition 11, there is a pair (x_1, x_2) and a coefficient $\lambda < 0$ or $\lambda > 1$ for which, for every z:

$$\lambda p(z|x_1) + (1-\lambda) p(z|x_2) \geq 0,$$

which implies that

$$\begin{aligned}\lambda p(y|x_1) + (1-\lambda)p(y|x_2) =& \lambda \int_z p(y|z)p(z|x_1)dz + (1-\lambda)\int_z p(y|z)p(z|x_2)dz\\ =& \int_z (\lambda p(z|x_1) + (1-\lambda)p(z|x_2))p(y|z)dz \geq 0.\end{aligned}$$

However, the latter cannot happen since $P(Y|X)$ is pairwise pure.

□

If both X and Z are binary, one can easily see that every non-deterministic relation between X and Z destroys pairwise purity. It is worth noticing that Theorem 6 is in practice applicable for low range Z. For large range of Z, the densities of $\{P(Z|x)\}_x$ can often be close to zero for some z-values. Then, non purity may not be detectable from empirical data.

So, according to Theorem 6, if a conditional $P(Y|X)$ is pairwise pure, then the existence of a low range variable Z, such that $X \perp\!\!\!\perp Y|Z$, is excluded. This implies that graphs belonging to the second category (see Section 6.1) are excluded. As a result, if a conditional $P(Y|X)$ is pairwise pure we conclude that $X \to Y$. Note, however, that non-pairwise-purity does *not* disprove that $X \to Y$ (see for example Lemma 7 in [Janzing et al., 2011]).

6.3 Empirical estimation of purity

To decide whether $P(Y|X)$ is pairwise pure, Lemma 4 is used. We employ kernel density estimation, using a Gaussian kernel, to estimate the conditional density $p(y|x)$ from finite data. Then, for all pairs (x, x'), we need to compute the minimum $\hat{p}(y|x)/\hat{p}(y|x')$ over y. Minimizing over all possible y is not feasible because the density estimate is unreliable in areas far from observed data points. Hence, we revert to a pragmatic solution, constraining $y \in \Psi$, with Ψ being a set of equally spaced points in the interval $[y_{\min}, y_{\max}]$, where $y_{\min}$ and $y_{\max}$ denote the minimum and the maximum of all observed y-values. That is, we compute $\min_{y\in\Psi}(\hat{p}(y|x)/\hat{p}(y|x'))$ for all pairs (x, x') and refer to the $\max_{\{(x,x')\}}(\min_{y\in\Psi}(\hat{p}(y|x)/\hat{p}(y|x')))$ as the *purity ratio* of the conditional $P(Y|X)$.

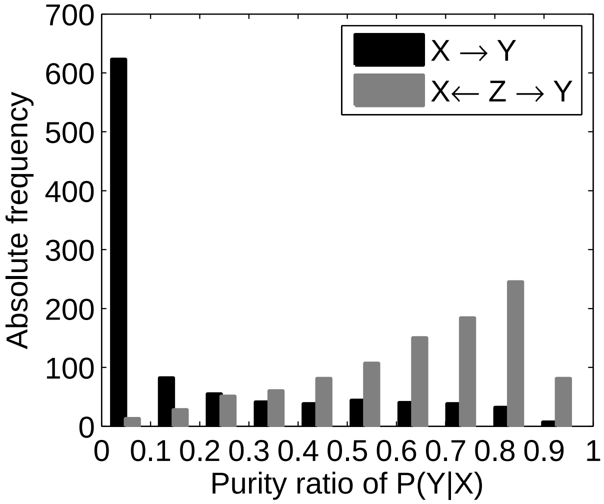


Figure 6.2: Histogram of the purity ratios for the two experimental settings; for the simulation setting $X \to Y$ the purity ratio values are closer to zero than for $X \leftarrow Z \to Y$.

6.4 Experiments

6.4.1 Simulated data

We consider variables X, Y, Z with ranges $\mathcal{X} := \{0, 1\}$, $\mathcal{Z} := \{0, 1\}$, $\mathcal{Y} := \mathbb{R}$, respectively. We first simulate the setting $X \to Y$. For that, we generate data according to a linear additive noise model $Y = wX + N$, with w a weight drawn from a zero mean Gaussian with unit variance. We choose N to be distributed according to a mixture of two Gaussians. Further, X is drawn from a Bernoulli distribution with success probability chosen uniformly at random from $[0, 1]$.

We then consider the setting $X \leftarrow Z \to Y$. We generate data using a linear additive noise model $Y = wZ + N$ with Z drawn from a Bernoulli distribution with success probability chosen uniformly at random from $[0, 1]$. The observed variable X is simulated using randomly chosen transition probabilities $P(X|Z)$. Specifically, $P(X = 0|Z = 0)$ and $P(X = 0|Z = 1)$ are drawn uniformly from $[0, 1]$.

We perform 1000 repetitions of each of the two experimental settings described above, drawing 1000 independent samples in each repetition. For each repetition we compute the purity ratio of $P(Y|X)$ (see Section 6.3). Figure 6.2 depicts the histogram of the estimated ratios for both settings. We can observe that purity ratios of $P(Y|X)$ from the first setting ($X \to Y$) are predominantly smaller than 0.1, whereas purity ratios of $P(Y|X)$ from the second setting ($X \leftarrow Z \to Y$) tend to yield higher values. Therefore, purity appears to be quite discriminative for these two settings.

6.4.2 Applications to statistical genetics

We next apply our method to a problem in statistical genetics, as already mentioned in the motivation of this work (Section 6.1). Reliable ground truth is difficult to obtain in genetic studies, and hence, following previous work (e.g., Platt et al. [2010]), we consider realistic simulated settings. Our simulation is based on data from a 250K SNP chip from *Arabidopsis*, consisting of 1200 samples (downloaded `http://walnut.usc.edu/2010/data/250k-data-version-3.06`). Hence, only the dependence between real genetic data and phenotype measurements is simulated, whereas the joint distribution of SNPs is based on real data.

Identifying causal SNPs using purity and correlation

We investigate to what extend the purity ratio is indicative of a causal relationship between a SNP and a phenotype. As a comparison, we also consider correlation, a basic measure of association that is commonly used in genetical studies [Balding, 2007].

We again consider two experimental settings. First, we simulate SNP-phenotype associations according to the setting $X \to Y$, first choosing a SNP X at random from the 250K SNPs, and then generating the phenotype Y from a linear additive noise model $Y = wX + N$ as before (see Section 6.4.1), where N here follows a Gaussian distribution.

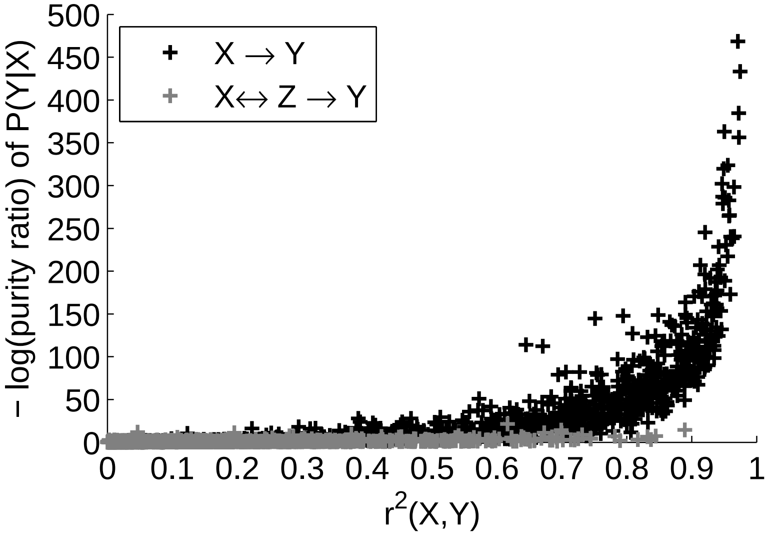


Figure 6.3: Scatter plot of the correlation between each SNP and its phenotype versus the negative logarithm of the purity ratio of $P(Y|X)$. Shown are SNPs that are causal (black) and non-causal (grey) separately. Even for strongly correlated non-causal SNPs, the negative logarithm of the purity ratio remains low and hence does not give false evidence for a causal link.

Analogously, we simulate associations according to the setting $X \leftrightarrow Z \rightarrow Y$. Here, Z is a SNP randomly selected among the set of all SNPs. We generate the phenotype Y, again according to a linear ANM, $Y = wZ + N$. The non-causal SNP X is chosen to be next to Z. This choice is motivated by the strong correlation between nearby SNPs, leading to an ambiguity as to which SNP is the causal one among a set of SNPs that may all exhibit strong correlation to the same phenotype.

In total, we generate 1000 (SNP X, phenotype Y) pairs according to each of the two settings described above, each pair consisting of 1200 samples as mentioned above. For each pair we estimate the purity ratio of $P(Y|X)$ as well as the correlation coefficient $r^2(X, Y)$. Fig. 6.3 shows the relationship between the correlation coefficients and the negative logarithm of the corresponding purity ratios, for both experimental settings. Notice that, high correlation coefficients are observed in both settings, while purity appears to discriminate between the settings. Even for strongly correlated non-causal

SNPs, the negative logarithm of the purity ratio remains low and hence does not give false evidence for a causal link $X \rightarrow Y$.

High correlation between the phenotype and the non-causal SNPs

Misleading correlation structure is a challenge in real association studies. A study in Platt et al. [2010] investigates very similar simulated models to highlight the risk of positively misleading answers from correlation analyses. We design this experiment such that the correlation between a non-causal SNP and its corresponding simulated phenotype can be higher than the correlation between the causal SNP and the phenotype.

We first simulate causal (SNP X, phenotype Y) pairs, generating Y as

$$Y = w_1 X + w_2 V + N \,,$$

where X is any random SNP, V is simulated as a corrupted version of another SNP located far from X and $w_2 = 2w_1$. Accordingly, we generate non-causal (X, Y) pairs, first choosing X randomly from the set of all SNPs and then generating Y as

$$Y = w_1 Z + w_2 V + N \,,$$

where Z is simulated as a corrupted version of X, V is a SNP located far from X and $w_1 = 2w_2$. To simulate a corrupted version of a SNP, we invert a certain percentage (corruption level) of its samples (here, $Z := X \oplus C$, with $\mathcal{C} := \{0, 1\}$ and $P(C = 1)$ being the corruption level).

Using the above setting for the weights of the models, we often get high correlations between simulated non-causal SNPs and their corresponding phenotypes and low correlations between simulated causal SNPs and their corresponding phenotype, which can be misleading for the inference of the causal direction. We compare the ability of purity and correlation to classify a SNP as causal or non-causal after generating 1000 causal SNP/phenotype and 1000 non-causal SNP/phenotype pairs. Fig. 6.4 shows the area under the receiver operating characteristic (ROC) curve (AUC) for both methods (purity and correlation) and for a range of different corruption levels. We can observe that purity consistently makes more accurate decisions than naive correlation analysis. In particular, for the limit of zero corruption both methods fail due to the strong coupling of non-causal SNPs with a simulated cause

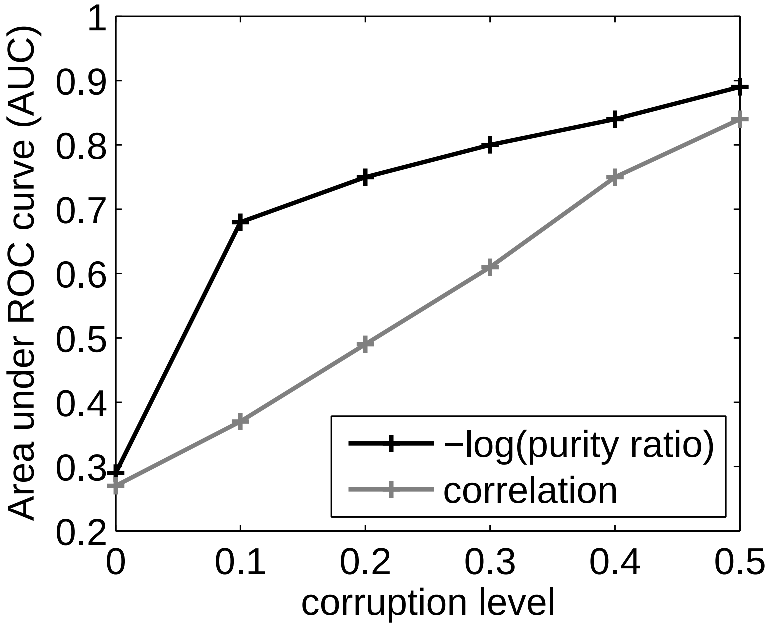


Figure 6.4: Area under the receiver operating characteristic curve (AUC) as a function of the corruption level, both using purity and correlation to identify a causal SNP. In the deterministic case (corruption=0) both methods fail, as non-causal SNPs are deterministically coupled with a simulated cause of Y. In the regime of high corruption levels (0.5), both methods perform equally well, since non-causal SNPs are not correlated anymore with the phenotype. In the relevant regime of an intermediate level of corruption, purity clearly outperforms the correlation measure.

of their corresponding Y: it is impossible to distinguish between too strongly coupled variables. In the regime of higher corruption, purity outperforms the correlation-based approach. Finally, in the limit of maximal corruption, both methods perform equally well, since the non-causal SNPs are not correlated anymore with the phenotype.

6.5 Conclusion

Motivated by a problem from statistical genetics, a method for causal discovery is proposed in this chapter that builds on a property of a conditional $P(Y|X)$, which we call purity. Purity is used as a criterion to infer $X \to Y$ as opposed to DAGs containing an unobserved low range variable Z in the path between X and Y that d-separates them. The characterization of a conditional as pure is based on the location of the different $\{P(Y|X=x)\}_x$ in the simplex of probability distributions of Y. We conducted experiments to estimate purity from finite data. The proposed method was found to perform better than standard correlation as of distinguishing cause-effect relations from spurious associations.

Chapter 7

Semi-supervised learning in causal and anticausal settings

7.1 Introduction

The motivation of this chapter is to study whether causal knowledge can be beneficial for traditional machine learning tasks like prediction problems. The goal of this chapter, as opposed to the previous ones, is not causal discovery. Instead, we argue that *knowing* the causal structure can have implications for semi-supervised-learning (SSL) tasks. Section 7.2 briefly describes SSL and the implication of causal knowledge for this task and Section 7.3 presents some empirical results.

7.2 SSL in causal and anticausal settings

In supervised learning we are given training data sampled from $P(X, Y)$. The goal is to learn a mapping from X to Y, i.e., to estimate $P(Y|X)$ (or properties thereof, e.g., its expectation). Then, the value of Y can be predicted for a new test value of X. X is called feature or predictor, while Y is called target. In case that the task is classification where Y is discrete, Y is often also called label or class.

In semi-supervised learning [Chapelle et al., 2006], the difference is that, apart from the samples drawn from $P(X, Y)$, we are given an additional set of unlabeled inputs from $P(X)$. In order to have a more accurate prediction by taking into account the unlabeled inputs, the distribution of the unlabeled data $P(X)$ has to carry information relevant for the estimation of $P(Y|X)$.

Consider first the task of SSL in case that we have the additional knowledge that the underlying causal structure is $X \to Y$. We call this the *causal setting* since we predict the effect Y from the cause X. Based on the principle of independence of causal mechanisms (see Postulate 1 in Section 4.1.4 and discussion thereafter), $P(X)$ contains no information about $P(Y|X)$. A more accurate estimate of $P(X)$, as may be possible by the addition of the unlabeled inputs from $P(X)$, does thus not influence an estimate of $P(Y|X)$, and SSL is pointless for this scenario.

Consider now the task of SSL in case $Y \to X$, which we call the *anticausal setting* since we predict the cause Y from the effect X. In this setting, the marginal distribution of the effect, $P(X)$, may contain information about $P(Y|X)$. The additional inputs from $P(X)$ may hence allow a more accurate estimate of $P(Y|X)$.

In conclusion, SSL is pointless in the causal setting (where $X \to Y$) but it may be helpful in the anticausal one (where $Y \to X$). The next section includes empirical results to support this hypothesis.

7.3 Empirical results

We do not perform new experiments. Instead, we check our hypothesis analyzing existing results of other papers. We compare the performance of SSL algorithms with that of base classifiers that use only labeled data.

For many datasets, X is vector-valued.[1] We first assign each dataset to one of the following three categories:[2]

[1] Only for this chapter we use the same notation X for both univariate and vector-valued variables. The rest of the chapters use bold face letters for vector-valued variables.

[2] The dataset categorization was performed in advance, before seeing the results of the meta-analysis, and was based on common sense.

1. *Anticausal/confounded:* (a) datasets in which at least one feature X_i is an effect of the target Y to be predicted (anticausal) (includes also cyclic causal relations between X_i and Y) and (b) datasets in which at least one predictor X_i has an unobserved common cause with the target Y to be predicted (confounded). In both (a) and (b) the mechanism $P(Y|X_i)$ can be dependent on $P(X_i)$. For these datasets, additional data from $P(X)$ may thus improve prediction.

2. *Causal:* datasets in which some predictors are causes of the target, and there is no predictor which (a) is an effect of the target or (b) has a common cause with the target. Based on the principle of independence of causal mechanisms, SSL should be futile on these datasets.

3. *Unclear:* datasets which are difficult to categorize into one of the aforementioned categories. Some of the reasons for that are incomplete documentation and lack of domain knowledge.

In practice, we count a dataset already as causal when we believe that the dependence between X and Y is *mainly* due to X causing Y, although additional confounding effects may be possible.

Table 7.1: Categorization of eight benchmark datasets as anticausal/confounded, causal or unclear

Category	Dataset	Reason of categorization
Anticausal/ confounded	g241c	The class causes the 241 features.
	g241d	The class (binary) and the features are confounded by a variable with four states.
	Digit1	The positive or negative angle and the features are confounded by the variable of continuous angle.
	USPS	The class and the features are confounded by the 10-state variable of all digits.
	COIL	The six-state class and the features are confounded by the 24-state variable of all objects.
Causal	SecStr	The amino acid is the cause of the secondary structure.
Unclear	BCI, Text	Unclear which is the cause and which the effect.

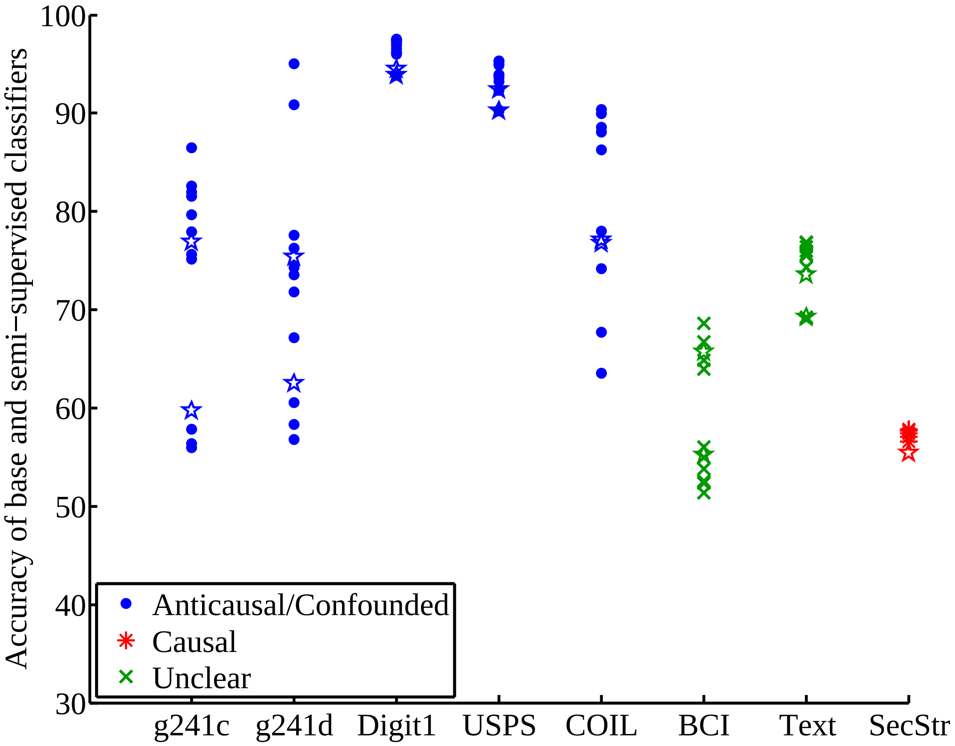


Figure 7.1: Accuracy of base classifiers (star shape) and different SSL methods on eight benchmark datasets.

7.3.1 Semi-supervised classification

We first analyze the results in the benchmark chapter of a book on SSL (Tables 21.11 and 21.13 of Chapelle et al. [2006]), for the case of 100 labeled training points. The chapter compares 11 SSL methods to the base classifiers 1-NN and SVM. In Table 7.1, we give details on our subjective categorization of the eight datasets used in the chapter.

In view of our hypothesis, it is encouraging to see (Fig. 7.1) that SSL does not significantly improve the accuracy in the one causal dataset, but it helps in most of the anticausal/confounded datasets. However, it is difficult to draw conclusions from this small collection of datasets; moreover, three additional issues may confound things: (1) the experiments were carried out

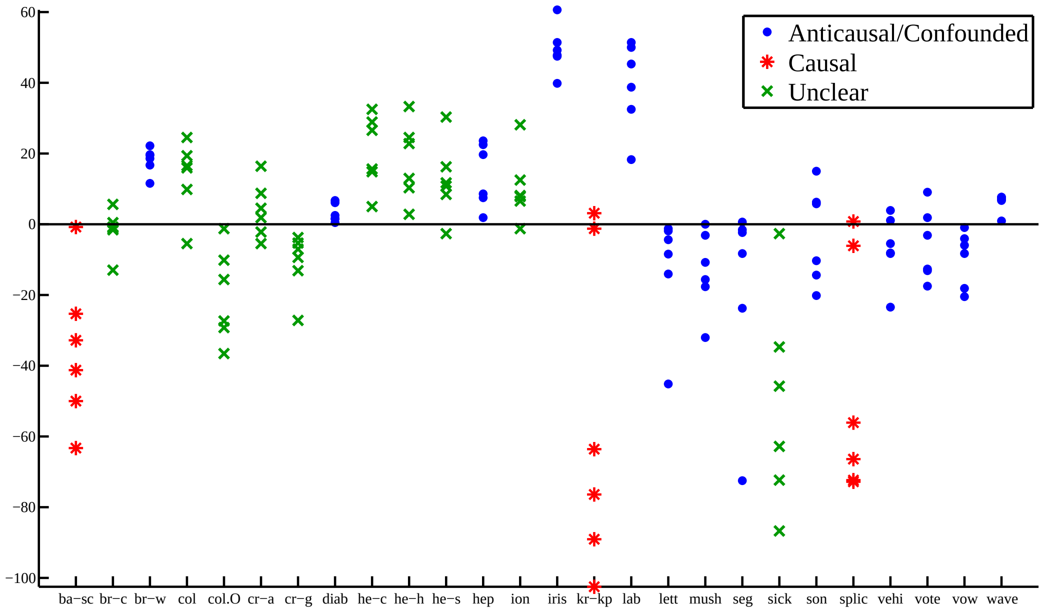


Figure 7.2: Plot of the relative decrease of error when using self-training, for six base classifiers on 26 UCI datasets. Here, relative decrease is defined as (error(base) − error(self-train)) / error(base). Self-training, a method for SSL, overall does not help for the causal datasets, but it does help for several of the anticausal/confounded datasets.

in a *transductive* setting. Inductive methods use labeled data to arrive at a classifier which is subsequently applied to an unknown test set; in contrast, transductive methods use the test inputs to make predictions. This could potentially allow performance improvements independent of whether a dataset is causal or anticausal; (2) the SSL methods used cover a broad range, and are not extensions of the base classifiers; (3) moreover, the results on the SecStr dataset are based on a different set of methods than the rest of the benchmarks.

We next consider 26 UCI datasets and six different base classifiers. The original results are from Tables III and IV in Guo et al. [2010], and are presently re-analyzed in terms of the above dataset categories. The comprehensive results of Guo et al. [2010] allow us the luxury of (1) considering only self-training, which is an extension of supervised learning to unlabeled data in the sense that if the set of unlabeled data is empty, we recover the results of the base method (in this case, self-training would stop at the first iteration). This lets us compare an SSL method to its corresponding base algorithm. Moreover, (2) we included only the *inductive* methods considered by Guo et al. [2010], and not the *transductive* ones (cf. our discussion above).

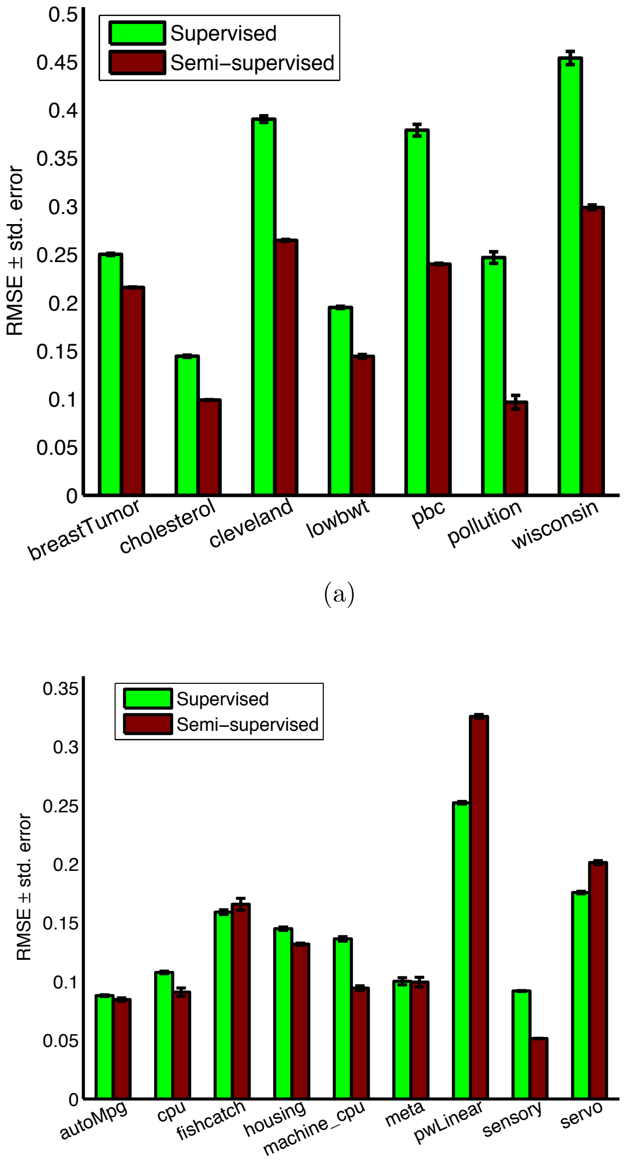
Table 7.2 describes our subjective categorization of the 26 UCI datasets into anticausal/confounded, causal, or unclear.

In Fig. 7.2, we observe that SSL does not significantly decrease the error rate in the three causal datasets, but it does increase the performance in several of the anticausal/confounded datasets. This is again consistent with our hypothesis that if mechanism and input are independent, SSL will not help for causal datasets.

7.3.2 Semi-supervised regression

Classification problems are often inherently asymmetric in that the inputs are continuous and the outputs categorical. It is worth reassuring that we obtain similar results in the case of semi-supervised regression (SSR). To this end, we consider the co-regularized least squares regression (co-RLSR) algorithm, compared to regular RLSR on 32 real-world datasets by Brefeld et al. [2006] (two of which are identical, so 31 datasets are considered). We categorize them into anticausal/confounded, causal, or unclear as in Table 7.3, prior to the subsequent analysis. Note that the categorization of Tables 7.2 and 7.3 is subjective and was made independently. That's the reason why the heart-c dataset (which coincides with the cleveland dataset) is categorized as unclear in Table 7.2 and as anticausal/confounded in Table 7.3. Nevertheless, this does not create any conflict with our claims.

We deem seven of the datasets anticausal, i.e., the target variable can be considered as the cause of (some of) the predictors; Fig. 7.3(a) shows that SSR reduces the root mean square errors (RMSEs) in all these cases. Nine of the remaining datasets can be considered causal, and Fig. 7.3(b) shows that there is usually little performance improvement for those. Like Brefeld et al. [2006], we use the Wilcoxon signed rank test to assess whether SSR outperforms supervised regression in the anticausal and causal cases. The null hypothesis is that the distribution of the difference between the RMSE



(a)

(b)

Figure 7.3: RMSE for (a) anticausal/confounded datasets and (b) causal datasets.

produced by SSR and that by supervised regression is symmetric around 0 (i.e., that SSR does not help). On the anticausal datasets, the p-value is 0.0156, while it is 0.6523 on the causal datasets. Therefore, we reject the null hypothesis in the anticausal case at a 5% significance level, but not in the causal case.

7.4 Conclusion

The aim of this chapter is to study whether causal knowledge can be beneficial for traditional machine learning tasks, specifically for semi-supervised learning. Our hypothesis is that SSL is pointless when predicting the effect from the cause, while it may be helpful when predicting the cause from the effect. The empirical results support this since the accuracy does not significantly improve for the causal datasets. A more rigorous analysis and understanding of the relation between the causal direction and the performance of SSL is left for future research.

Table 7.2: Categorization of 26 UCI datasets as anticausal/confounded, causal or unclear

Categ.	Dataset	Reason of categorization
Anticausal/confounded	breast-w	The class of the tumor (benign or malignant) causes some of the features of the tumor (e.g., thickness, size, shape).
	diabetes	Whether or not a person has diabetes affects some of the features (e.g., glucose concentration, blood pressure), but is also an effect of some others (e.g., age, number of times pregnant).
	hepatitis	The class (die or survive) and many of the features (e.g., fatigue, anorexia, liver big) are confounded by the presence or absence of hepatitis. Some of the features, however, may also cause death.
	iris	The size of the plant is an effect of the category it belongs.
	labor	Cyclic causal relationships: good or bad labor relations can cause or be caused by many features (e.g., wage increase, number of working hours per week, number of paid vacation days, employer's help during employee 's long term disability). Moreover, the features and the class may be confounded by elements of the character of the employer and the employee (e.g., ability to cooperate).
	letter	The class (letter) is a cause of the produced image of the letter.
	mushroom	The attributes of the mushroom (shape, size) and the class (edible or poisonous) are confounded by the taxonomy of the mushroom (23 species).
	segment	The class of the image is the cause of its features.
	sonar	The class (Mine or Rock) causes the sonar signals.
	vehicle	The class of the vehicle causes the features of its silhouette.
	vote	This dataset may contain causal, anticausal, confounded and cyclic causal relations. E.g., having handicapped infants or being part of religious groups in school can cause one's vote, being democrat or republican can causally influence whether one supports Nicaraguan contras, immigration may have a cyclic causal relation with the class. Crime and the class may be confounded, e.g., by the environment in which one grew up.
	vowel	The class (vowel) causes the features.
	waveform-5000	The class of the wave causes its attributes.
Causal	balance-scale	The features (weight and distance) cause the class.
	kr-vs-kp	The board-description influences whether white will win.
	splice	The DNA sequence causes the splice sites.
Unclear	breast-cancer, colic, colic.ORIG, credit-a, credit-g, heart-c, heart-h, heart-statlog, ionosphere, sick	In some of the datasets, it is unclear whether the class label may have been generated or defined based on the features (e.g., ionoshpere, credit, sick).

Table 7.3: Categorization of 31 UCI datasets as anticausal/confounded, causal or unclear

	Dataset	Target variable	Reason of categorization
Anticausal/confounded	breastTumor	tumor size	causing predictors such as inv-nodes and deg-malig
	cholesterol	cholesterol	causing predictors such as resting blood pressure and fasting blood sugar
	cleveland	presence of heart disease in the patient	causing predictors such as chest pain type, resting blood pressure, and fasting blood sugar
	lowbwt	birth weight	causing the predictor indicating low birth weight
	pbc	histologic stage of disease	causing predictors such as Serum bilirubin, Prothrombin time, and Albumin
	pollution	age-adjusted mortality rate per 100,000	causing the predictor number of 1960 SMSA population aged 65 or older
	wisconsin	time to recur of breast cancer	causing predictors such as perimeter, smoothness, and concavity
Causal	autoMpg	city-cycle fuel consumption in miles per gallon	caused by predictors such as horsepower and weight
	cpu	cpu relative performance	caused by predictors such as machine cycle time, maximum main memory, and cache memory
	fishcatch	fish weight	caused by predictors such as fish length and fish width
	housing	housing values in suburbs of Boston	caused by predictors such as pupil-teacher ratio and nitric oxides concentration
	machine_cpu	cpu relative performance	see remark on “cpu”
	meta	normalized prediction error	caused by predictors such as number of examples, number of attributes, and entropy of classes
	pwLinear	value of piecewise linear function	caused by all 10 involved predictors
	sensory	wine quality	caused by predictors such as trellis
	servo	rise time of a servomechanism	caused by predictors such as gain settings and choices of mechanical linkages
Unclear	auto93 (target: midrange price of cars); bodyfat (target: percentage of body fat); autoHorse (target: price of cars); autoPrice (target: price of cars); baskball (target: points scored per minute); cloud (target: period rainfalls in the east target); echoMonths (target: number of months patient survived); fruitfly (target: longevity of mail fruitflies); pharynx (target: patient survival); pyrim (quantitative structure activity relationships); sleep (target: total sleep in hours per day); stock (target: price of one particular stock); strike (target: strike volume); triazines (target: activity); veteran (survival in days)		

Chapter 8

Inference of cause and effect with unsupervised inverse regression

8.1 Introduction

The goal of this chapter is to solve Problem 2 (causal structure learning) for the case of only two variables, i.e., $\mathbf{X} = (X, Y)$. Specifically, the task is to decide between $X \rightarrow Y$ and $Y \rightarrow X$ (assuming no confounders) for two continuous univariate random variables X and Y, given a sample from their joint distribution, $P(X, Y)$. We assume that $P(X, Y)$ has a density $p_{X,Y}(x, y)$ with respect to the Lebesgue measure.

We employ the principle of independence of causal mechanisms (Postulate 1). As discussed in Section 4.1.4, for deterministic non-linear relations, Janzing et al. [2012] and Daniusis et al. [2010] define independence through uncorrelatedness between $\log f'$ and p_X, both viewed as random variables. For non-deterministic relations, considered in this chapter, it is not obvious how to explicitly formalize independence between $P(X)$ and $P(Y|X)$. Motivated by the previous chapter, we propose an implicit notion of independence, namely that $P(Y|X)$ cannot be estimated based on $P(X)$. However, it may be possible to estimate $P(X|Y)$ based on $P(Y)$.

In Chapter 7 we argued that *knowing* the causal direction has implications for semi-supervised learning. Specifically, if $X \to Y$, $P(X)$ contains no information about $P(Y|X)$ according to Postulate 1. As a result, a more accurate estimate of $P(X)$, as may be possible by the addition of the extra unlabeled points in SSL, does not influence an estimate of $P(Y|X)$, and SSL is pointless in this scenario. In contrast, SSL may be helpful in case $Y \to X$. Thus, the notion of independence between $P(X)$ and $P(Y|X)$ implicitly reads: the former is not helpful for estimating the latter.

The use of Postulate 1 in this chapter complies with the latter notion of independence: if $X \to Y$, estimating $P(Y|X)$ based on $P(X)$ should not be possible. In contrast, estimating $P(X|Y)$ given $P(Y)$ may be possible. Employing this asymmetry, we propose CURE (Causal discovery with Unsupervised inverse REgression), a method to infer the causal graph in case of two variables, that is appropriate for non-deterministic relations. The proposed causal discovery method infers $X \to Y$ if the estimation of $P(X|Y)$ based on $P(Y)$ is more accurate than the one of $P(Y|X)$ based on $P(X)$. Otherwise, $Y \to X$ is inferred.

To this end, we propose a method for estimating a conditional distribution based on samples from the corresponding marginal. We call it unsupervised inverse GP regression for the following reason: in standard supervised regression, given a sample from $P(X, Y)$, the goal is to estimate the conditional $P(Y|X)$. We call supervised *inverse* regression the task of estimating the conditional $P(X|Y)$, without changing the original regression model of Y on X that was used for the estimation of $P(Y|X)$. Our task is to estimate the conditional $P(X|Y)$ based only on samples from the marginal $P(Y)$. We, thus, call it *unsupervised inverse regression.*

Sections 8.2 and 8.3.2 describe the building blocks for unsupervised inverse regression, presented in Section 8.3.1. Section 8.4 then describes CURE, which applies unsupervised inverse regression two times, one for each direction, and infers the causal direction by comparing the resulting estimations of the conditionals $P(X|Y)$ and $P(Y|X)$. We evaluate CURE on synthetic and real data (Section 8.6). On the latter, our method outperforms existing causal inference methods.

8.2 Gaussian process latent variable model

The Gaussian process latent variable model (GP-LVM) [Lawrence, 2005] can be interpreted as a multi-output Gaussian process (GP) model [Rasmussen and Williams, 2006] in which only the output data are observed, while the input remain latent. Let $\mathbf{y} \in \mathbb{R}^{n\times d}$ be the observed data where n is the number of observations and d the dimensionality of each observation. Further, let $\mathbf{x} \in \mathbb{R}^{n\times p}$ denote the unobserved input data. The purpose is often dimensionality reduction, thus $p \ll d$. GP-LVM defines a mapping from the latent to the observed space by using GPs, with hyperparameters $\boldsymbol{\theta}$. Assuming independence across the dimensions, the likelihood function is given as:

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \prod_{j=1}^{d} p(\mathbf{y}_j|\mathbf{x},\boldsymbol{\theta})$$

where $\mathbf{y}_j$ the j^{th} column of $\mathbf{y}$,

$$p(\mathbf{y}_j|\mathbf{x},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_j; \mathbf{0}, K_{\mathbf{x},\mathbf{x}} + \sigma^2 I_n),$$

and $K_{\mathbf{x},\mathbf{x}}$ the $n \times n$ covariance function defined by a selected kernel function. Thus, $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$ is a product of d independent Gaussian processes where the input, $\mathbf{x}$, is latent.

For the present work, only univariate random variables are relevant, thus $d = p = 1$. This defines a *single-output* GP-LVM, i.e., just one GP model with latent input. In this case, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$ and the likelihood function of single-output GP-LVM is given as:

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{\mathbf{x},\mathbf{x}} + \sigma^2 I_n) \tag{8.1}$$

with $\boldsymbol{\theta} = (\ell, \sigma_f, \sigma)$, where we choose the RBF kernel

$$k(x^{(i)}, x^{(j)}) = \{K_{\mathbf{x},\mathbf{x}}\}_{i,j} = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\left(x^{(i)} - x^{(j)}\right)^2\right)$$

with $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})$.

Lawrence [2005] finds $\mathbf{x}$ (for multiple-output GP-LVM), by MAP estimation, selecting a Gaussian prior for $\mathbf{x}$, while jointly maximizing with respect to $\boldsymbol{\theta}$. In Bayesian GP-LVM [Titsias and Lawrence, 2010], instead, $\mathbf{x}$ is variationally integrated out and a lower bound on the marginal likelihood $p(\mathbf{y})$ is computed.

8.3 Unsupervised inverse regression

As mentioned in the introduction of this chapter, if $X \to Y$, estimating $P(X|Y)$ based on the distribution of the effect, $P(Y)$, may be possible. In this section we present a method to accomplish this. Throughout the rest of the chapter, let $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ be a sample of n independently and identically distributed (i.i.d.) observations from $P(X, Y)$. Further let $\mathbf{x} := (x^{(1)}, \ldots, x^{(n)})$ and $\mathbf{y} := (y^{(1)}, \ldots, y^{(n)})$. Moreover, $\mathbf{x}$ and $\mathbf{y}$ are rescaled between zero and one.

The goal of this section is to estimate $p_{X|Y}$ based on $\mathbf{y}$.

8.3.1 Unsupervised inverse GP Regression

Since the estimation of the conditional is based *only* on samples $\mathbf{y}$ from the marginal $P(Y)$, X is considered *latent*. A Gaussian process regression model of Y on X is used, which can be alternatively seen as single-output GP-LVM described in Section 8.2. Specifically, according to Eq. (8.1):

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{\mathbf{x},\mathbf{x}} + \sigma^2 I_n)$$

Further, a uniform prior, $\mathcal{U}(0,1)$, is chosen for the distribution of X. A uniform prior is, additionally, placed over $\boldsymbol{\theta}$ which suppresses overly flexible functions (small ℓ) to restrict the function class.

Using the aforementioned model, we estimate $p_{X|Y}$ based on $\mathbf{y}$ by

$$\hat{p}^{\mathbf{y}}_{X|Y} : (x, y) \mapsto p(x|y, \mathbf{y}).$$

The predictive distribution $p(x|y, \mathbf{y})$ is given by marginalizing over the latent n-dimensional vector $\mathbf{x}$ and the *unknown* GP hyperparameters $\boldsymbol{\theta}$:

$$\begin{aligned} p(x|y, \mathbf{y}) &= \int_{\mathcal{X},\Theta} p(\mathbf{x}, \boldsymbol{\theta}, x|\mathbf{y}, y) d\mathbf{x} d\boldsymbol{\theta} \\ &= \int_{\mathcal{X},\Theta} p(x|y, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}, y) d\mathbf{x} d\boldsymbol{\theta} \end{aligned}$$

$$\approx \int_{\mathcal{X},\Theta} p(x|y, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) d\mathbf{x} d\boldsymbol{\theta} \tag{8.2}$$

The first factor, $p(x|y, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta})$, is the predictive distribution of *supervised inverse* GP regression, which is described in Section 8.3.2 (Eq. (8.5)). The second factor, $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$, is the posterior distribution over $\mathbf{x}$ and the hyperparameters $\boldsymbol{\theta}$, given the observed $\mathbf{y}$.

By Bayes' theorem:

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) p(\boldsymbol{\theta})}{p(\mathbf{y})} \propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) p(\boldsymbol{\theta}) \\ &= p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{\mathbf{x},\mathbf{x}} + \sigma^2 I_n) \end{aligned} \tag{8.3}$$

Note that the computation of the latent's posterior distribution $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ is analytically intractable since $\mathbf{x}$ appears non-linearly inside the inverse of $K_{\mathbf{x},\mathbf{x}} + \sigma^2 I_n$ [Titsias and Lawrence, 2010]. In our implementation, we approximate the posterior $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ using a Markov Chain Monte Carlo (MCMC) method, slice sampling [Neal, 2003]. The sample size n determines the dimensionality of the space to sample from, which is $n + 3$ (including the three hyperparameters). Thus, the computational complexity is determined by n and this step poses the main computational bottleneck of our algorithm.

$p(x|y, \mathbf{y})$ is then estimated by replacing the integral in (8.2) with a sum over m MCMC samples from $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$:

$$p(x|y, \mathbf{y}) \approx \frac{1}{m} \sum_{i=1}^{m} p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i) \tag{8.4}$$

So, $p(x|y, \mathbf{y})$ is computed as the average of predictive distributions of supervised inverse regressions. Each predictive distribution $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$ uses the i^{th} sample, $(\mathbf{x}_i, \boldsymbol{\theta}_i)$, from the posterior $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$.

8.3.2 Supervised inverse GP Regression

Following from the previous section, the remaining task is to compute $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$ in (Eq. (8.4)), for each MCMC sample i, with $1 \leq i \leq m$.

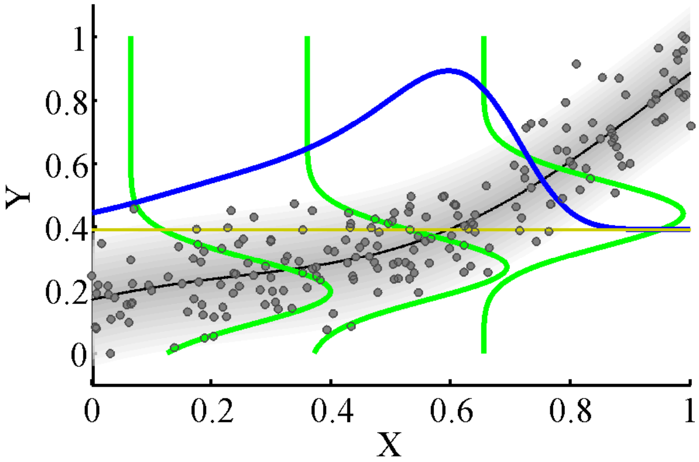


Figure 8.1: The predictive distributions of standard GP regression at three x values (green) and the predictive distribution of supervised inverse GP regression at one y value (blue). The latter is not Gaussian.

Since X is independent of the hyperparameters and the distribution of X is uniform, by Bayes' theorem we get:

$$\begin{aligned} p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i) &\propto \; p(\mathbf{y}, y|\mathbf{x}_i, x, \boldsymbol{\theta}_i) p(x|\mathbf{x}_i, \boldsymbol{\theta}_i) \\ &= \mathcal{N}(\mathbf{y}, y; \mathbf{0}, K_{(\mathbf{x}_i,x),(\mathbf{x}_i,x)} + {\sigma_i}^2 I_n) \end{aligned} \tag{8.5}$$

Notice that, unlike standard GP regression, the predictive distribution of inverse GP regression, $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$, is not Gaussian (for fixed y). We first compute $\mathcal{N}(\mathbf{y}, y; \mathbf{0}, K_{(\mathbf{x}_i,x),(\mathbf{x}_i,x)} + {\sigma_i}^2 I_n)$ at the points of a grid on the range of X, $[0, 1]$, and then normalize appropriately to get $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$. Fig. 8.1 illustrates an example of supervised inverse regression. The predictive distributions of standard GP regression, $p(y|x, \mathbf{x}_i, \mathbf{y}, \boldsymbol{\theta}_i)$, (for some i) at three x values are depicted in green and the predictive distribution of inverse GP regression, $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$, at one y value (yellow line), in blue.

The usual practice to estimate $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$ would be to learn directly a map from Y to X (discriminative model). However, we need to use GP regression of Y on X and not of X on Y in order to comply with the model used in Section 8.3.1.

To conclude, $p(x|y, \mathbf{y})$ is computed from Eq. (8.4), using Eq. (8.5) for each

$p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$. Likewise, we can compute $p(y|x, \mathbf{x})$ repeating the above procedure with a GP regression model of X on Y.

8.3.3 Evaluation

In Sections 8.3.1 and 8.3.2 we proposed a method to estimate $p_{X|Y}$ based on $\mathbf{y}$, by

$$\hat{p}^{\mathbf{y}}_{X|Y} : (x, y) \mapsto p(x|y, \mathbf{y}),$$

with $p(x|y, \mathbf{y})$ computed from Eq. (8.4), using Eq. (8.5). In this section we evaluate the accuracy of our estimation of $p_{X|Y}$. We compute the negative log likelihood $L^{\text{unsup}}_{X|Y} = -\sum_{i=1}^{n} \log \hat{p}^{\mathbf{y}}_{X|Y}(x^{(i)}, y^{(i)})$ at $\mathbf{x}$, $\mathbf{y}$ to measure the performance of unsupervised inverse regression. We could also evaluate it at new test points if we had a separate test set $\mathbf{x}_{\text{te}}$, $\mathbf{y}_{\text{te}}$. However, since the task is unsupervised, we do not have overfitting issues and use all data for estimating $p_{X|Y}$. In order to evaluate the accuracy of the estimation of $p_{X|Y}$, we compare $L^{\text{unsup}}_{X|Y}$ with the accuracy of the corresponding supervised inverse regression $L^{\text{sup}}_{X|Y} = -\sum_{i=1}^{n} \log \hat{p}^{\mathbf{x},\mathbf{y}}_{X|Y}(x^{(i)}, y^{(i)})$, using again a uniform prior for X but with $\boldsymbol{\theta}$ computed by maximization of $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$. Specifically,

$$\hat{p}^{\mathbf{x},\mathbf{y}}_{X|Y} : (x, y) \mapsto p(x|y, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}),$$

with $p(x|y, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta})$ obtained according to Eq. (8.5). This way, we measure how much the performance degrades due to the absence of $\mathbf{x}$, specifically by:

$$D_{X|Y} = L^{\text{unsup}}_{X|Y} - L^{\text{sup}}_{X|Y} \tag{8.6}$$

8.4 CURE

The ultimate goal of this chapter is to decide upon the causal direction, $X \to Y$ or $Y \to X$, given $\mathbf{x}$ and $\mathbf{y}$. According to Postulate 1, if $X \to Y$, estimating $P(Y|X)$ from $P(X)$ should not be possible. In contrast, estimating $P(X|Y)$ based on $P(Y)$ may be possible. So, CURE is given as follows: if we can estimate $P(X|Y)$ based on samples from $P(Y)$ more accurately than $P(Y|X)$ based on samples from $P(X)$, then $X \to Y$ is inferred. Otherwise, $Y \to X$ is inferred. In particular, we apply unsupervised inverse GP regression two times. First, $D_{X|Y}$ is computed as in (8.6):

$$D_{X|Y} = L^{\text{unsup}}_{X|Y} - L^{\text{sup}}_{X|Y} = -\sum_{i=1}^{n} \log \hat{p}^{\mathbf{y}}_{X|Y}(x^{(i)}, y^{(i)}) + \sum_{i=1}^{n} \log \hat{p}^{\mathbf{x},\mathbf{y}}_{X|Y}(x^{(i)}, y^{(i)})$$

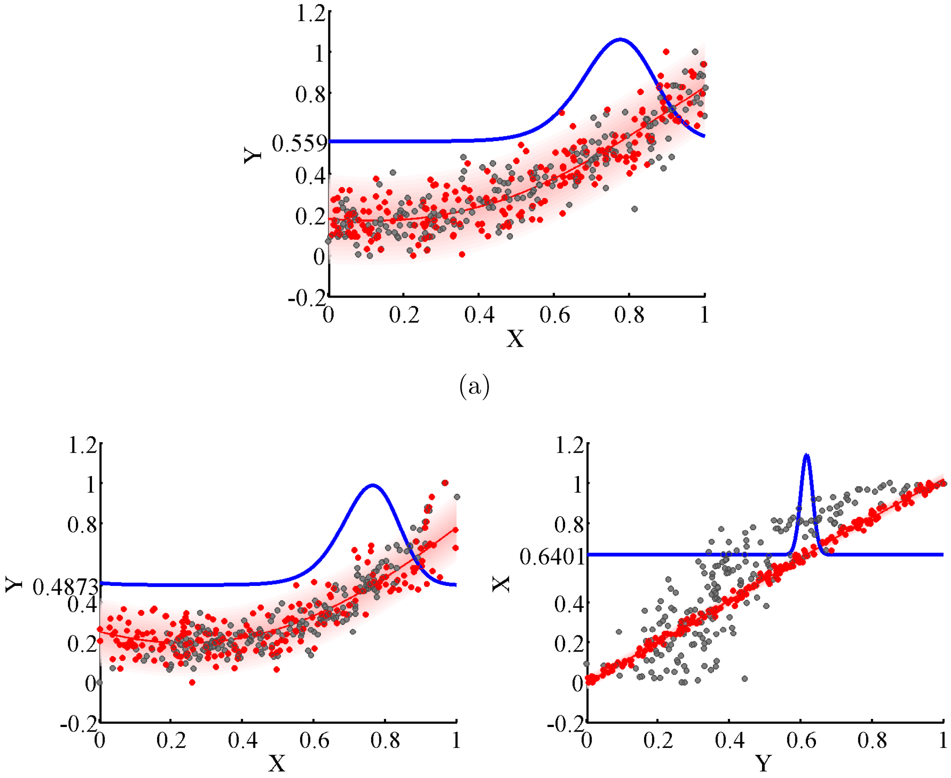
to evaluate the estimation of $p_{X|Y}$ based on $\mathbf{y}$. Then, $D_{Y|X}$ is computed as:

$$D_{Y|X} = L^{\text{unsup}}_{Y|X} - L^{\text{sup}}_{Y|X} = -\sum_{i=1}^{n} \log \hat{p}^{\mathbf{x}}_{Y|X}(y^{(i)}, x^{(i)}) + \sum_{i=1}^{n} \log \hat{p}^{\mathbf{y},\mathbf{x}}_{Y|X}(y^{(i)}, x^{(i)})$$

to evaluate the estimation of $p_{Y|X}$ based on $\mathbf{x}$. Finally, we compare the two performances: if $D_{X|Y} < D_{Y|X}$, then we infer the causal direction to be $X \to Y$, otherwise we output $Y \to X$.

8.5 Discussion

Figure 8.2 depicts three datasets generated according to causal models with DAG $X \to Y$ (grey points) (note the exchanged axes in the last figure). In particular, in Figs. 8.2(a) and 8.2(c) the grey points are generated according to $Y = 2X^3 + X + E$, with X having a uniform distribution and E zero-mean Gaussian noise. On the other hand, the distribution of X in Fig. 8.2(b) is sub-Gaussian and the noise is not additive. Since $X \to Y$, we expect to be able to estimate $P(X|Y)$ based on $P(Y)$ more accurately than $P(Y|X)$ based on $P(X)$. The quality of the estimation strongly depends on the generated MCMC samples from the high-dimensional posterior in Eq. (8.3). Figures 8.2(a) and 8.2(b) refer to the estimation of $P(X|Y)$ based on samples from $P(Y)$, whereas Fig. 8.2(c) to the estimation of $P(Y|X)$ based on samples from $P(X)$. In Figs. 8.2(a) and 8.2(b) the y-coordinates of the red points correspond to $\mathbf{y}$ and the x-coordinates to *one* MCMC sample from $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ (Eq. (8.3)). Given the sample $(\mathbf{x}_i, \boldsymbol{\theta}_i)$, $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$, plotted in blue for a fixed y, is computed by supervised inverse GP regression. We can observe (Fig. 8.2(b)) that even when data are not generated according to the simple model used in Section 8.3.1 we often still get relatively “good” MCMC samples.



(a)

(b)

(c)

Figure 8.2: The grey points are generated according to $X \rightarrow Y$. (a), (c): uniform $P(X)$, additive Gaussian noise, (b): sub-Gaussian $P(X)$, non-additive noise. (a), (b): the y-coordinates of the red points correspond to $\mathbf{y}$ and the x-coordinates to one MCMC sample from $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$. Given the sample $(\mathbf{x}_i, \boldsymbol{\theta}_i)$, $p(x|y, \mathbf{y}, \mathbf{x}_i, \boldsymbol{\theta}_i)$, plotted in blue, is computed by supervised inverse GP regression. (c): note that x and y axes are exchanged. The x-coordinates of the red points correspond to $\mathbf{x}$ and the y-coordinates to one sample from $p(\mathbf{y}, \boldsymbol{\theta}|\mathbf{x})$. Given the sample $(\mathbf{y}_i, \boldsymbol{\theta}_i)$, $p(y|x = 0.64, \mathbf{x}, \mathbf{y}_i, \boldsymbol{\theta}_i)$, plotted in blue, is computed by inverse regression. In (b) we see that a relatively "good" MCMC sample is obtained when using only the distribution of the effect, even when the distribution of the cause is not uniform and the noise is not additive. In contrast, when using only the distribution of the cause, we often get "bad" MCMC samples, like the one depicted in red in (c).

On the contrary, in Fig. 8.2(c) the x-coordinates of the red points correspond to $\mathbf{x}$ and the y-coordinates to one MCMC sample from $p(\mathbf{y}, \boldsymbol{\theta}|\mathbf{x})$. In this case we often get “bad” MCMC samples as expected since we should not be able to estimate $P(Y|X)$ based on samples from $P(X)$ (Postulate 1). So, even in cases where the (unrealistic) model assumptions of Section 8.3.1 do not hold, the estimation of $P(X|Y)$ is often still better than that of $P(Y|X)$. That is, even though the estimation of $P(X|Y)$ can be far from the true $P(X|Y)$, the estimation of $P(Y|X)$ can be even further from the true $P(Y|X)$. Proving such a claim, however, is not trivial, but the experiments are encouraging in this direction.

The step of sampling from the high dimensional distribution $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ is not trivial. Additionally, there are two modes with equal probabilities, namely, one that corresponds to the ground truth $\mathbf{x}$ and one to the “mirror” of $\mathbf{x}$ (flipping $\mathbf{x}$ left to right). Good initialization is crucial for sampling from this high-dimensional space. The good news is that, for the purpose of causal inference, we have the luxury of initializing the sampling algorithm with the ground truth $\mathbf{x}$, since this is given (but we treat it as a latent variable), and with hyperparameters computed by maximizing the likelihood $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$. This is fair as long as it is done for both causal directions to be checked. With this initialization, slice sampling starts from the correct mode of $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ and usually (apart from very noisy cases), we do not get samples from the “mirror” mode. In any case, for every sample, $\mathbf{x}$ is used to decide to keep either this or its mirror. Initializing slice sampling with $\mathbf{x}$, we still get an asymmetry between cause and effect: even by initializing with the ground truth $\mathbf{x}$, if $Y \to X$ and we try to predict $P(X|Y)$ from $P(Y)$ (which are independent), then we eventually often get “bad” MCMC samples similar to the one in Fig. 8.2(c). Of course, this slice sampling initialization is only feasible for the purpose of causal inference, where both $\mathbf{x}$ and $\mathbf{y}$ are given. If the goal is just estimating $P(X|Y)$ based on samples from $P(Y)$, then we only get to see $\mathbf{y}$ and such a sampling initialization is not possible. In that sense, to be precise, the conditional $P(X|Y)$ is not estimated based only on $\mathbf{y}$, but also using some side information for $\mathbf{x}$ (for sampling initialization).

One final point of discussion is the choice of the hyperparameters’ prior. Non-invertible functional relationships between the observed variables can provide clues to the generating causal model [Friedman and Nachman, 2000]. In contrast, in the invertible case it gets more difficult to infer the causal direction. This is one more reason to restrict $\boldsymbol{\theta}$ to favor more regular functions (of large length-scale).

8.6 Experiments

8.6.1 Simulated data

We generate data both with additive noise, according to $Y = f(X) + N$, with $f(X) = bX^3 + X$, and non-additive noise. Non-additive noise is simulated according to $Y = f(X) + N$, with $P(N) = \sigma\mathcal{N}(0,1)\,|\sin(2\pi\nu X)| + \frac{1}{4}\sigma\mathcal{N}(0,1)\,|\sin(2\pi(10\nu)X)|$.[1] By multiplying with a sinusoidal function the width of the noise varies for different values of X. The parameter ν controls the frequency of the wave. The results are included in Fig. 8.3, for a non-linear f (setting $b = 2$), and in Fig. 8.4, for a linear f (setting $b = 0$). The three first columns of the figures refer to data generated with additive noise and the fourth column with non-additive noise. We use four distributions for $P(X)$: standard uniform, sub-Gaussian, Gaussian and super-Gaussian, each one corresponding to one row of Figs. 8.3 and 8.4. For sub- and super-Gaussian, data are generated from a Gaussian distribution and their absolute values are raised to the power q while keeping the original sign. $q = 0.7$ for the sub-Gaussian distribution (which is also close to bimodal), while $q = 1.3$ for the super-Gaussian. Similarly, three distributions are used for $P(N)$: sub-Gaussian, Gaussian, and super-Gaussian, each one corresponding to one of the first three columns of Figs. 8.3 and 8.4. The x-axis of the first three columns refers to the standard deviation (std) of the noise. Three values of std are used: $0.25, 0.45$ and 0.8, each multiplied by the standard deviation of $f(X)$ in order to get comparable results across different experiments. The x-axis of the fourth column is the frequency of the sinusoidal wave, ν, with values from $\{4, 0.5, 025\}$. We generate $n = 200$ samples for each simulated setting.

We compare the proposed causal inference method (CURE) with some of

[1] Note that we call $Y = f(X) + N$ an additive noise model only if $X \perp\!\!\!\perp N$. This comes from the perspective of structural equations where the noise term is independent of X. Then, a conditional $P(Y|X)$ generated by *dependent additive* noise can only be generated by a structural equation with non-additive noise.

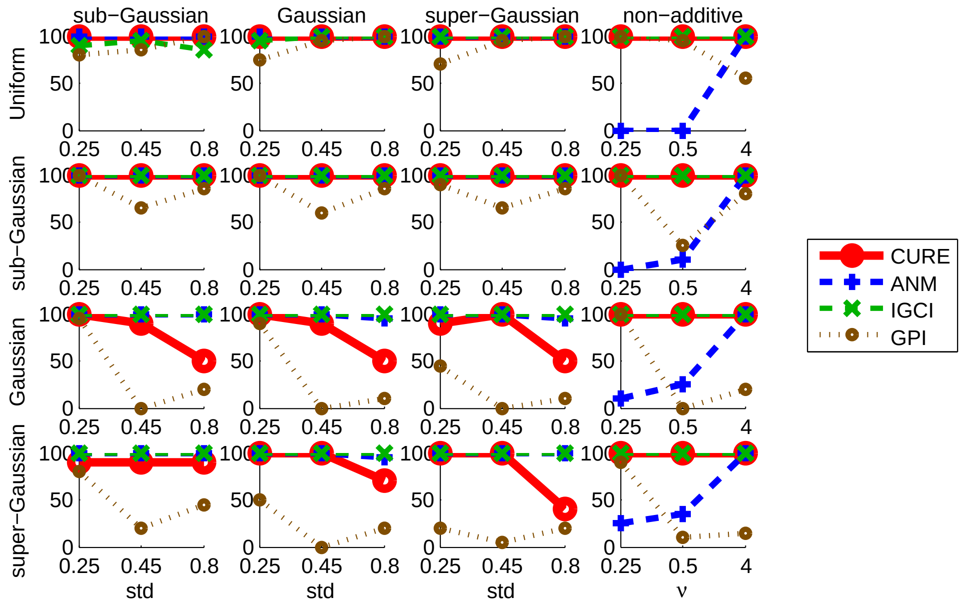


Figure 8.3: Performance (percentage of correct causal inferences) of various causal inference methods for simulated data with a non-linear function f. Rows correspond to the distribution of the cause, $P(X)$. The three first columns correspond to the distribution, $P(N)$, of the additive noise term, with the x-axis referring to 3 different standard deviations of the noise. The last column corresponds to non-additive noise, with the x-axis referring to 3 different frequencies of the sinusoidal wave (used to simulate non-additive noise).

the causal inference methods reviewed in Chapter 4: additive noise models (ANMs) [Hoyer et al., 2009, Peters et al., 2014], information-geometric causal inference (IGCI) [Daniusis et al., 2010, Janzing et al., 2012] and Bayesian model selection (GPI) [Mooij et al., 2010]. CURE uses a uniform prior so a preprocessing step is first applied to X and Y to remove possible isolated points (low-density points). For CURE, $m = 15000$ MCMC samples are generated from the 203-dimensional ($n = 200$) posterior using the slice sampling method, from which the first 5000 are discarded. Since it is difficult to sample from this very high-dimensional space, to get a more robust answer, we report the average $D_{X|Y}$ and $D_{Y|X}$ across 4 repetitions of CURE for each

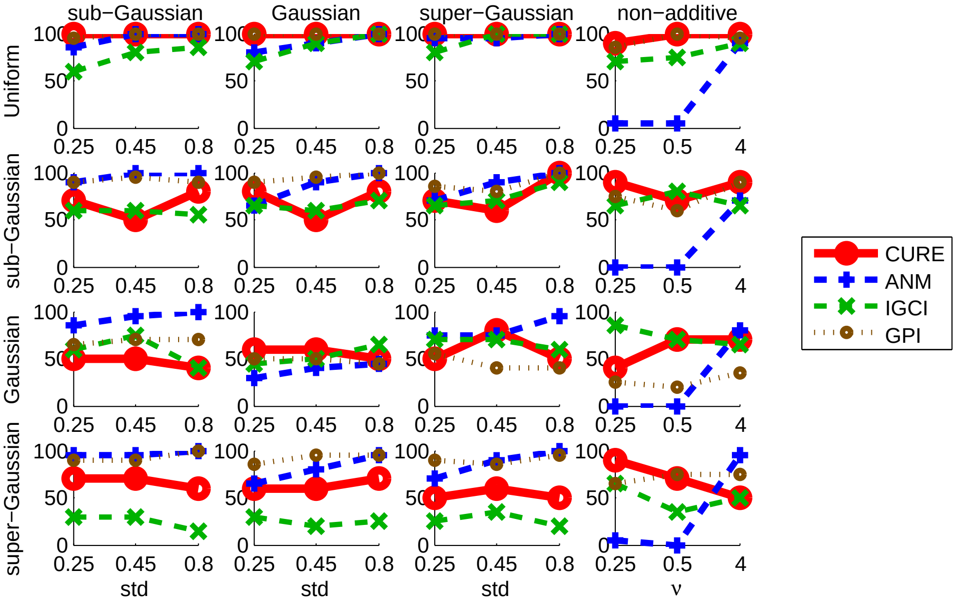


Figure 8.4: As in Fig. 8.3 but with a linear function f.

dataset. We call those repetitions "internal" repetitions of the CURE algorithm to distinguish them from the repetitions of the simulations. Assume $D^i_{X|Y}$ is the output of the i^{th} internal repetition. Then, $D_{X|Y} = \frac{1}{4}\sum_{i=1}^{4} D^i_{X|Y}$ and $D_{Y|X} = \frac{1}{4}\sum_{i=1}^{4} D^i_{Y|X}$. We conduct 20 repetitions for each combination of method and simulation setting, apart from CURE which is repeated 10 times, due to the high computational complexity of the MCMC sampling step. The y-axis of Figs. 8.3 and 8.4 corresponds to the percentage of correct causal inferences.

For non-linear f (Fig. 8.3), we can observe that CURE (red) infers correctly the causal direction when $P(X)$ is uniform or sub-Gaussian and for all noise distributions. The accuracy degrades in some cases of Gaussian and super-Gaussian $P(X)$ (due to the uniform prior) with high standard deviation of $P(N)$. IGCI (green) infers the causal direction correctly in almost all cases, even though it was proposed for deterministic relations. ANM (blue) gets 100% correct decisions on the additive noise data, however, its performance really degrades when it comes to non-additive noise. Finally, GPI (brown) performs better with uniform $P(X)$ than with Gaussian or super-Gaussian, where its results are worse compared to the other methods.

For the linear case (Fig. 8.4), the performance of almost all methods gets worse since it gets more difficult to recover the causal direction. Specifically, the case of linear f and Gaussian $P(X)$ and $P(N)$ is non-identifiable [Hoyer et al., 2009]. This is also supported by the results: in this case the decision of all methods is close to 50% (random guess). For uniform $P(X)$, CURE outperforms the other methods, however for non-uniform $P(X)$ its performance often degrades. ANM generally performs relatively well with additive noise, however, it again fails in the non-additive noise case. GPI performs much better in the linear compared to the non-linear case, outperforming the other methods in several cases. Finally, IGCI often fails in the linear case.

8.6.2 Real data

Further, we evaluate the performance of our method on real-world data, namely on the database with cause-effect pairs[2] (version 0.9) [Mooij et al., 2014]. It consists of 86 pairs of variables from various domains with known causal structure, the first 41 of which are from the UCI Machine Learning Repository [Bache and Lichman, 2013]. The task is to infer the causal direction for each of the pairs. Each pair is weighted as suggested in the database. Five of the pairs have multivariate X or Y and are excluded from the analysis. At most $n = 200$ samples from each cause-effect pair are used (less than 200 only if the pair itself has less samples). For CURE, $m = 10000$ MCMC samples are generated, after burning the first 10000 samples and additionally discarding every other sample. The average $D_{X|Y}$ and $D_{Y|X}$ across 8 internal repetitions of CURE are computed for each dataset. Two more methods (introduced in Section 4.1.3) participate in this comparison: Post-Nonlinear Models (PNL) [Zhang and Hyvärinen, 2009] and Linear Non-Gaussian Acyclic Models (LiNGAM) [Shimizu et al., 2006]. The results for all the methods are depicted in Fig. 8.5. The y-axis corresponds to the percentage of correct causal inferences. As the causal inference methods we compare with, we also output a ranking of the pairs according to a confidence criterion along with the decisions on the causal direction. The method

[2] http://webdav.tuebingen.mpg.de/cause-effect/

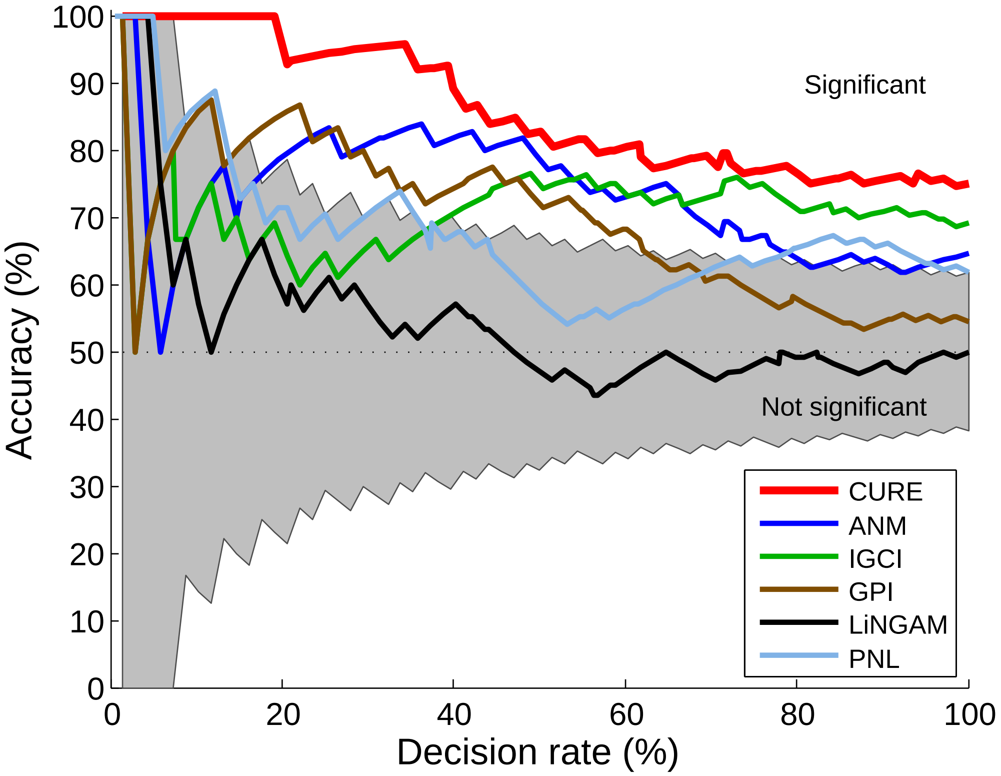


Figure 8.5: Results of various causal inference methods for 81 cause-effect pairs (86 excluding 5 multivariate pairs), showing the percentage of correct causal inferences for each decision rate.

is more certain about the decided direction of the top-ranked pairs as opposed to the low-ranked ones. Using this ranking, we can decide on the causal direction of only a subset of the pairs for which we are more confident about. This way, we trade off accuracy versus the number of decisions taken. The x-axis of Fig. 8.5 corresponds to the percentage of pairs for which we infer the causal direction (100% means that we are forced to decide upon the direction of all 81 pairs). A good confidence criterion corresponds to the accuracy being lowest for decision rate 100% and increase monotonically as the decision rate decreases. As a confidence criterion we choose to use the ratio between $\sigma_{D_{X|Y}} = std(\{D^i_{X|Y}\}_{1\leq i\leq 8})$ and $\sigma_{D_{Y|X}} = std(\{D^i_{Y|X}\}_{1\leq i\leq 8})$, with denominator the one that corresponds to the inferred causal direction (smaller D). The idea is that, if $X \to Y$ and we try to predict $P(X|Y)$ based on $P(Y)$, the empirical variance of the algorithm across internal repetitions is expected to be small: MCMC samples are expected to correspond to conditionals close to the ones of the ground truth. On the other hand, when predicting $P(Y|X)$ based on $P(X)$ (which are independent), the variance is higher across internal repetitions.

We consider the null hypothesis that "the causal inference algorithm outputs random decisions with probability 1/2 each". Then the grey area of Fig. 8.5 indicates the 95% confidence interval of a binomial distribution with k trials where k is the (weighted) number of cause-effect pairs (the weights given as suggested in the database). Thus, the area outside the grey area corresponds to results significantly correlated with the ground truth. We can observe that CURE (bold red) outperforms the other methods for all decision rates, however it is difficult to draw any definite conclusions about the relative performance of these methods based on only 81 cause-effect pairs. Moreover, the ratio of standard deviations that is used as a confidence criterion for CURE seems to be a good choice: for low decision rates we even get 100% accuracy, decreasing more or less monotonically as the decision rate increases. IGCI performs well for high decision rates but its confidence criterion does not behave as expected. ANM has a better confidence criterion, however, its performance is quite low compared to CURE and IGCI when it is forced to take a decision. The result of PNL is marginally significant in the forced-decision regime. Finally, the results of GPI and LINGAM are not significantly correlated with the ground truth in the forced-decision regime.

Increasing n, the performance is obviously increasing. For example, running ANM with all the available samples of the 81 cause-effect pairs results in an accuracy of 72% [Peters et al., 2014], much higher than its accuracy with $n = 200$ (Fig. 8.5). Unfortunately, the computational complexity of CURE does not allow for it to be run for such a big sample size (thousands for some pairs). However, we consider very encouraging the fact that CURE can yield accuracy 75% already with $n = 200$.

8.7 Conclusion

We propose a method (CURE) to infer the causal direction between two random variables given a sample from their joint distribution. It is based on the principle of independence of causal mechanisms (Postulate 1). If we can estimate $P(X|Y)$ based on $P(Y)$ more accurately than $P(Y|X)$ based on $P(X)$, then $X \to Y$, is inferred. Otherwise, $Y \to X$ is inferred. For that, unsupervised inverse GP regression is proposed as a method to estimate a conditional from samples from the corresponding marginal. CURE was evaluated both on simulated and real data, and found to perform well compared to existing methods. In particular, it outperforms five existing causal inference methods on our real data experiments. A downside is the comparably high computational cost due to the large number of required MCMC steps.

Chapter 9

Empirical performance of ANMs

This chapter is concerned with the empirical behavior of estimation methods for causal discovery using Additive Noise Models (ANMs) [Hoyer et al., 2009, Peters et al., 2014]. We focus on the two-variable case, i.e., $\mathbf{X} = (X, Y)$. Existing methods (see Section 4.1.3) have proven identifiability of ANMs: if $Y = f(X) + N_Y$ with $X \perp\!\!\!\perp N_Y$, then, in the generic case, there is no function g and noise variable N_X such that $X = g(Y) + N_X$, with $Y \perp\!\!\!\perp N_X$.

The structure learning algorithm then reads: whenever there is an ANM in one direction inducing the joint distribution $P(X, Y)$, but there is no ANM in the other direction inducing $P(X, Y)$, then the DAG corresponding to the former direction is inferred (for more details see Section 4.1.3). In particular, causal discovery under ANM is performed according to the following procedure [Hoyer et al., 2009]:

1. Regress Y on X and obtain the residuals $N_{Y,f} = Y - f(X)$.
2. Regress X on Y and obtain the residuals $N_{X,g} = X - g(Y)$.
3. Infer $X \to Y$ if $N_{Y,f} \perp\!\!\!\perp X$ but $N_{X,g} \not\!\perp\!\!\!\perp Y$. Decide $Y \to X$ if the reverse holds true, abstain otherwise.

Instantiations of the above procedure vary in the regression methods employed for function fitting and in the independence measures employed. Hoyer et al. [2009] perform the regression using Gaussian Processes [Rasmussen and Williams, 2006] and the independence tests using the Hilbert Schmidt Independence Criterion (HSIC) [Gretton et al., 2008]. We employ two different regression methods: kernel regression (KR) and kernel ridge regression (KRR). The following lemma is used to explain the measure of independence that we use.

Let H and I denote differential entropy and mutual information, respectively [Cover et al., 1994].

Lemma 5

$$H(X) + H(N_{Y,f}) = H(Y) + H(N_{X,g}) - I(N_{X,g}, Y) - I(N_{Y,f}, X).$$

Proof. By the chain rule of differential entropy we have

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) = H(X) + H(N_{Y,f}|X) \\ &= H(X) + H(N_{Y,f}) - I(N_{Y,f}, X), \text{ similarly} \end{aligned}$$

$$H(X,Y) = H(Y) + H(N_{X,g}) - I(N_{X,g}, Y).$$

Equate the two r.h.s. above and rearrange. □

Note that whenever $N_{Y,f} \perp\!\!\!\perp X$, we have $I(N_{Y,f}, X) = 0$. Therefore, by the above lemma, if $N_{Y,f} \perp\!\!\!\perp X$ then $C_{XY} := H(X) + H(N_{Y,f})$ is smaller than $C_{YX} := H(Y) + H(N_{X,g})$. So, the third step of the above procedure becomes:

3. Infer $X \rightarrow Y$ if $C_{YX} - C_{XY} > 0$. Decide $Y \rightarrow X$ if $C_{YX} - C_{XY} < 0$, abstain otherwise.

This yields a measure of independence which is relatively cheap to estimate. In particular the test depends only on the marginal distributions of X, Y and residuals, and does not involve estimating joint distributions or conditionals, as is implicit in most independence tests.

In practice we are given a finite sample $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ drawn i.i.d. from $P(X,Y)$. Then, let f_n denote an estimate for the regression function f and g_n an estimate for g. Further, let N_{Y,f_n} and N_{X,g_n} the corresponding residuals. Let H_n denote an entropy estimator. For entropy estimation we employ a resubstitution estimate using a kernel density estimator [Beirlant et al., 1997].

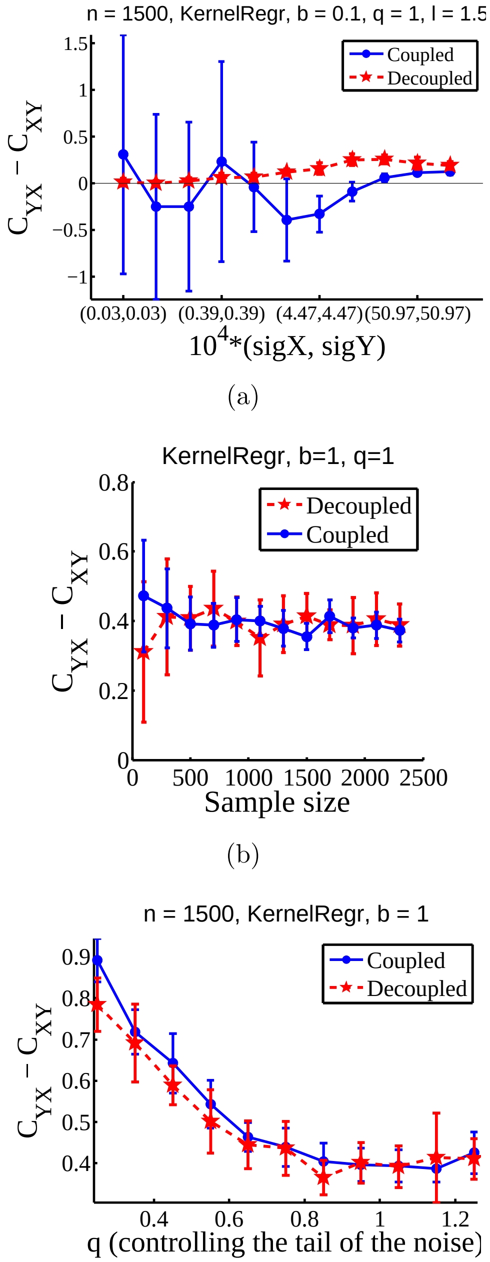
We consider two different estimation scenarios described below.

Definition 12 (Decoupled estimation) *f_n and g_n are learned on half of the sample $\{(x^{(i)}, y^{(i)})\}_{1 \leq i \leq n}$, and the $H_n(N_{Y,f_n})$ and $H_n(N_{X,g_n})$ are learned on the other half of the sample (w.l.o.g. assume n is even). $H_n(X)$ and $H_n(Y)$ could be learned on either half or on the entire sample.*

Definition 13 (Coupled estimation) *All f_n, g_n and entropies H_n are learned on the entire sample $\{(x^{(i)}, y^{(i)})\}_{1 \leq i \leq n}$.*

We present a series of experimental results. Specifically, in all our experiments, simulated data are generated as $Y = bX^3 + X + N$. X is sampled from a uniform distribution on the interval $[-2.5, 2.5]$, while N is sampled as $|\mathcal{N}|^q \cdot \mathrm{sign}(\mathcal{N})$ where $\mathcal{N}$ is a standard normal. The parameter b controls the strength of the nonlinearity of the function while q controls the non-Gaussianity of the noise: $q = 1$ gives a Gaussian, while $q > 1$ and $q < 1$ produces super-Gaussian and sub-Gaussian distributions, respectively.

In each of the plots of Figs. 9.1 and 9.2 the y-axis is the estimated difference $(C_{YX} - C_{XY})$. Moreover, coupled and decoupled estimation are overlayed, illustrated by blue and red colors, respectively. Figure 9.1 concerns using kernel regression as opposed to Fig. 9.2 which includes results using kernel ridge regression. In Fig. 9.1, $(C_{YX} - C_{XY})$ is plotted against varying the richness of the regression algorithm (Fig. 9.1(a)), the sample size (Fig. 9.1(b)) and the tail of the noise (Fig. 9.1(c)). Specifically, we control the richness of the algorithm by varying the kernel bandwidth of regressor and the tail of the noise by varying q. The experiments of Figs. 9.1(b) and 9.1(c) are repeated using kernel ridge regression and the results are depicted in Figs. 9.2(a) and 9.2(b), respectively. In all figures, apart from Fig. 9.1(a), the kernel bandwidth of the regressor is tuned by cross-validation. For every combination of the parameters, each experiment is repeated 10 times and average results for $(C_{YX} - C_{XY})$ are reported along with standard deviation across repetitions.


n = 1500, KernelRegr, b = 0.1, q = 1, l = 1.5
Coupled
Decoupled
C_YX − C_XY
1.5
1
0.5
0
−0.5
−1
(0.03,0.03)
(0.39,0.39)
(4.47,4.47)
(50.97,50.97)
10^4*(sigX, sigY)
KernelRegr, b=1, q=1
Decoupled
Coupled
C_YX − C_XY
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
Sample size
n = 1500, KernelRegr, b = 1
Coupled
Decoupled
C_YX − C_XY
0.9
0.8
0.7
0.6
0.5
0.4
0.4
0.6
0.8
1
1.2
q (controlling the tail of the noise)

(a)

(b)

(c)

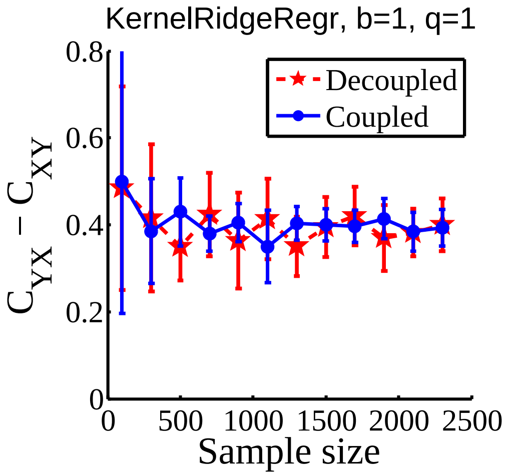
Figure 9.1 *(previous page)*: Plots of the difference between the complexity measures, $(C_{YX} - C_{XY})$, for coupled and decoupled estimation in various scenarios using kernel regression (KR). For every combination of the parameters, each experiment is repeated 10 times and average results for $(C_{YX} - C_{XY})$ are reported along with standard deviation across repetitions. (a): increasing kernel bandwidth of regressor geometrically (by factors of $l = 1.5$), i.e. decreasing richness of the algorithm. When the capacity of the regression algorithm is too large, the variance of the causal inference is large for coupled estimation (due to overfitting) but remains low for decoupled estimation. (b): increasing sample size. For tuned bandwidth, the variance of the causal inference is only due to the sample size, so the coupled estimation, which estimates everything on a larger sample, becomes the better procedure. (c): increasing q, i.e. the tail of the noise is made sharper. For faster decreasing tail of the noise, the causal inference performs better.

By decoupling regression and entropy estimations, we reduce the potential of overfitting, during entropy estimation, the generalization error of regression. This generalization error could be large if the regression algorithms are too rich. Our simulations show that, when the regression algorithm is too rich, the variance of the causal inference is large for coupled estimation but remains low for decoupled estimation (Fig. 9.1(a)). By decreasing the richness of the class (simulated by increasing the kernel bandwidth for a kernel regressor) the source of variance shifts to the sample size, and coupled estimation (which estimates everything on a larger sample) becomes the better procedure and tends to converge faster (Fig. 9.1(b)).

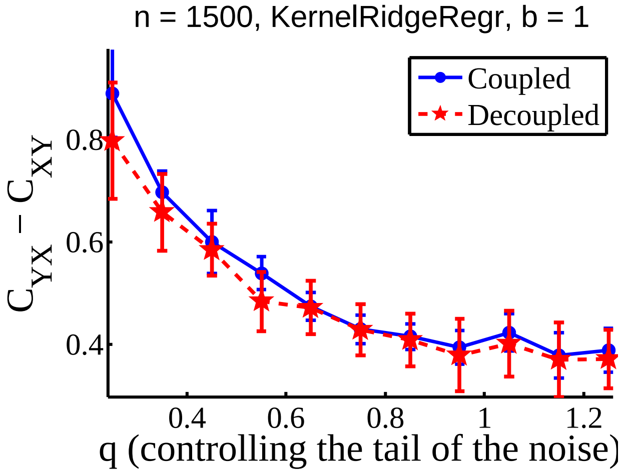
In Kpotufe et al. [2014], we suggest that convergence of causal inference is likely faster if the noise has faster decreasing tail. This is verified in our simulations where we vary the tail of the noise (Fig. 9.1(c)).

Comparing Figs. 9.1 and 9.2, the selection of regression method does not seem to matter for the causal inference results.

Mooij et al. [2014] recently also presented some empirical results on the performance of ANMs on real and simulated data, focusing on different aspects such as model misspecification.



(a)



(b)

Figure 9.2: (a) Same experiment as Fig. 9.1(b) but using KRR and (b) same experiment as Fig. 9.1(c) but using KRR. For properly tuned parameters, the selection of regression method does not seem to matter for the causal inference results.

Chapter 10

Conclusions and future work

This thesis introduces several novel, mostly nonparametric, methods for causal discovery from observational data. In this chapter we provide a brief summary of them before proposing possible directions for future research. Chapters 5 and 6 are concerned with causal discovery when allowing for confounders while Chapters 8 and 9 are devoted to causal discovery in the two-variable case, assuming no confounders. In Chapter 7 we argue that causal knowledge can be useful for standard machine learning tasks, such as semi-supervised learning.

In particular, in Chapter 5 we present a method for identification of finite mixtures of product distributions. The proposed method is further used for identifying confounders. Chapter 6, motivated by an application in genetics, introduces a property of a conditional distribution, called purity. Using pure conditionals we are able to exclude the existence of a low range unobserved variable that d-separates two observed ones.

Chapters 7 and 8 are based on the principle of independence of causal mechanisms. Chapter 8 proposes CURE, a causal discovery method for the two-variable case. The method suggests to infer the causal direction by comparing the estimations of the conditionals based on the corresponding marginals in both directions. Chapter 7 argues that SSL is meaningful only in the anticausal setting, where the target is the cause and the feature the effect. On the contrary, in the causal setting SSL is pointless. Finally, Chapter 9 presents empirical results concerning the behavior of estimation methods for causal discovery using ANMs in the two-variable case.

There are several open questions for future research:

Principle of independence of causal mechanisms. Chapters 7 and 8 are using Postulate 1. Future research should concentrate on being more explicit as to what is meant by *independence* or *information.* As mentioned in Section 4.1.4, Janzing and Schölkopf [2010] postulate *algorithmic* independence of $P(Y|X)$ and $P(X)$, i.e. zero algorithmic mutual information. This is equivalent to saying that the shortest description (in the sense of Kolmogorov complexity) of $P(X, Y)$ is given by separate descriptions $P(X)$ and $P(Y|X)$. Since Kolmogorov complexity is uncomputable, practical implementations must rely on other notions of (in)dependence or information. For deterministic non-linear relations, Janzing et al. [2012] and Daniusis et al. [2010] define independence through uncorrelatedness between $\log f'$ and the density of $P(X)$ w.r.t. the Lebesgue measure. In the non-deterministic case there is room for future research on providing formalizations of Postulate 1 and specifying what kind of information the conditional shares with the marginal.

CURE. CURE, proposed in Chapter 8, argues that, according to Postulate 1, if $X \to Y$, estimating $P(Y|X)$ based on $P(X)$ should not be possible. In contrast, estimating $P(X|Y)$ given $P(Y)$ may be possible. The proposed causal discovery method exploits this asymmetry and infers $X \to Y$ if the estimation of $P(X|Y)$ based on $P(Y)$ is more accurate than the one of $P(Y|X)$ based on $P(X)$. Otherwise, $Y \to X$ is inferred. Future work should focus on better understanding what kind of information is used to estimate the conditional from the corresponding marginal distribution of the effect and for which distributions the asymmetry is expected to hold. For example, it would be interesting to understand under what conditions or assumptions the simple model used in Section 8.3.1 is guaranteed to better infer the conditional when based on the marginal of the effect as opposed to the marginal of the cause.

Hypothesis test for purity. It would be interesting to develop a hypothesis test to decide when to reject the null hypothesis of a conditional being pure.

Larger networks. The causal discovery methods presented in this thesis

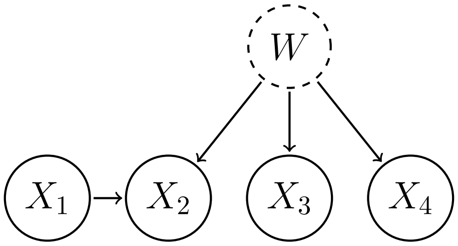


Figure 10.1: Example of showing how to extend the method proposed in Chapter 5. The rank of $\mathcal{U}_{X_1,X_2,X_3,X_4}$ is infinite, whereas the rank of $\mathcal{U}_{X_1,X_2,X_3}$ is finite.

concern mostly simple networks such as the two-variable case (e.g., Chapters 8 or 9) or networks with at most one latent variable (e.g., Chapter 5). Future research should focus on the generalization of these ideas to larger networks. One simple extension, for example, could be the following: in the method of Chapter 5, instead of taking into account only the rank of the Hilbert space embedding of the joint distribution of *all* observed variables, $\mathcal{U}_{\mathbf{X}}$, consider also the rank of the embedding of the joint distribution of *subsets* of $\mathbf{X}$. Combining these partial informations, one could arrive at more conclusions. Consider, for example, the DAG G of Fig. 10.1. The current algorithm would not arrive at any conclusions since the rank of $\mathcal{U}_{X_1,X_2,X_3,X_4}$ is infinite.[1] However, with the added extension of also checking the rank of the subsets, the finite rank of $\mathcal{U}_{X_1,X_2,X_3}$ would lead to the detection and reconstruction of the confounder W.

Real data. Future work should also focus on conducting more real data experiments. Ultimately, we should aim at developing causal discovery methods applicable to large scale real applications from various domains, such as biology, medicine, finance etc. To this end, assumptions that are more realistic for practical applications should be considered.

[1] Assuming, according to Section 5.6, that the $\{X_j\}$ are continuous, W has a small number of states and $(G, P(X_1, X_2, X_3, X_4, W))$ is a full rank BN.

Bibliography

E. S. Allman, C. Matias, and J. A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37:3099–3132, 2009.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

D. J. Balding. *Handbook of statistical genetics*. Wiley-Interscience, 2007.

J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. Van der Meulen. Nonparametric entropy estimation: an overview. *International Journal of Mathematical and Statistical Sciences*, 6:17–40, 1997.

M. Benaglia, D. Chauveau, and D. R. Hunter. Bandwidth selection in an EM-like algorithm for nonparametric multivariate mixtures. *In Nonparametrics and Mixture Models*, 2011.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. Böhning and W. Seidel. Editorial: recent developments in mixture models. *Computational Statistics & Data Analysis*, 41(3):349–357, 2003.

U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *Proceedings of the 23th International Conference on Machine Learning (ICML)*, pages 137–144, 2006.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

D. Chauveau, D. R. Hunter, and M. Levine. Estimation for conditional independence multivariate finite mixture models. *Technical Report*, 2010.

D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

T. Claassen and T. Heskes. A Bayesian approach to constraint based causal inference. In *Proceedings of the 28th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 207–216, 2012.

T. Claassen, J. Mooij, and T. Heskes. Learning sparse causal models is not NP-hard. In *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 172–181, 2013.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

T. M. Cover, J. A. Thomas, and J. Kieffer. Elements of information theory. *SIAM Review*, 36(3):509–510, 1994.

P. Daniusis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. In *Proceedings of the 26th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

V. De Silva and L.H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

F. Eberhardt and R. Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

Z. D. Feng and C. E. McCulloch. Using bootstrap likelihood ratios in finite mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(3):609–617, 1996.

N. Friedman and I. Nachman. Gaussian process networks. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–219, 2000.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 489–496. MIT Press, 09 2008.

D. Geiger and D. Heckerman. Learning Gaussian networks. In *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 235–243, 1994.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems (NIPS) 20*, pages 585–592, Cambridge, 2008. MIT Press.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:671 –721, 2012.

Y. Guo, X. Niu, and H. Zhang. An extensive empirical study on semi-supervised learning. In *IEEE International Conference on Data Mining (ICDM)*, pages 186–195, 2010.

P. Hall and X.-H. Zhou. Nonparametric estimation of component distributions in a multivariate mixture. *The Annals of Statistics*, 31(1):201–224, 2003.

D. Heckerman. A bayesian approach to learning causal networks. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 285–295, 1995.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20 (3):197–243, 1995.

P. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 689–696, 2009.

Y. Huang and M. Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 217–224, 2006.

T. Iwata, D. Duvenaud, and Z. Ghahramani. Warped mixtures for nonparametric cluster shapes. In *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

D. Janzing and B. Steudel. Justifying additive noise model-based causal discovery via algorithmic information theory. *Open Systems & Information Dynamics*, 17(02):189–212, 2010.

D. Janzing, J. Peters, J. Mooij, and B. Schölkopf. Identifying latent confounders using additive noise models. In *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 249–257, 2009.

D. Janzing, E. Sgouritsa, O. Stegle, J. Peters, and B. Schölkopf. Detecting low-complexity unobserved causes. In *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 383–391, 2011.

D. Janzing, J. M. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniusis, B. Steudel, and B. Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182-183:1–31, 2012.

H. Kasahara and K. Shimotsu. Nonparametric identification of multivariate mixtures. *Journal of the Royal Statistical Society-Series B*, 2010.

K. Korb, L. Hope, A. Nicholson, and K. Axnick. Varieties of causal intervention. In *Proceedings of the Pacific Rim Conference on AI*, pages 322–331, 2004.

S. Kpotufe, E. Sgouritsa, D. Janzing, and B. Schölkopf. Consistency of causal inference under the additive noise model. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 478–486, 2014.

J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95 – 138, 1977.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

J. Lemeire and E. Dirkx. Causal models as minimal descriptions of multivariate systems. `http://parallel.vub.ac.be/∼jan/`, 2006.

M. Levine, D. R. Hunter, and D. Chauveau. Maximum smoothed likelihood for multivariate mixtures. *Biometrika*, 98(2):403–416, 2011.

J. M. Mooij, O. Stegle, D. Janzing, K. Zhang, and B. Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 1687–1695, 2010.

J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *arXiv:1412.3773*, 2014.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT press, 2012.

R. M. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, 1988.

J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2nd edition, 2009.

J. Peters. *Restricted Structural Equation Models for Causal Inference.* PhD thesis, ETH Zurich and MPI Tübingen, 2012.

J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

A. Platt, B. J. Vilhjalmsson, and M. Nordborg. Conditions under which genome-wide association studies will be positively misleading. *Genetics*, 186(3):1045, 2010.

C. E. Rasmussen. The infinite gaussian mixture model. *Advances in Neural Information Processing Systems (NIPS)*, 12(5.2):2, 2000.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

T. Richardson and P. Spirtes. Ancestral graph markov models. *Annals of Statistics*, pages 962–1030, 2002.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1255–1262, 2012.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. J. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

S. Shimizu, P. Hoyer, and A. Hyvärinen. Estimation of linear non-gaussian acyclic models for latent factors. *Neurocomputing*, 72(7–9):2024–2027, 2009.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, volume 21, pages 1219–1226, 2006.

R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7: 191–246, 2006.

A. Smola, A. Gretton, L. Song, and B. Schölkopf. A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer-Verlag, 2007.

P. Spirtes. Introduction to causal inference. *Journal of Machine Learning Research*, 11:1643–1662, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

J. Tian and J. Pearl. Causal discovery from changes. In *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence (UAI)*, pages 512–521, 2001.

J. Tian and I. Shpitser. On identifying causal effects. In H. Dechter, R. Geffner and J. Halpern, editors, *Heuristics, Probability and Causality: A Tribute to Judea Pearl.*, pages 415–444. College Publications, 2010.

M. Titsias and N. Lawrence. Bayesian gaussian process latent variable model. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 844–851, 2010.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1): 31–78, 2006.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 220–227, 1991.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 647–655, 2009.

Appendix A

Causal sufficiency example

This Appendix presents an example that explains the difference between the two definitions of causal sufficiency included in Section 2.3. Specifically, the set $\{X, Y\}$ of the DAG in Fig. A.1 is causally sufficient according to Definition 7 but not according to the standard definition of causal sufficiency, e.g., [Spirtes, 2010], since Z is an unobserved common cause of X and Y.

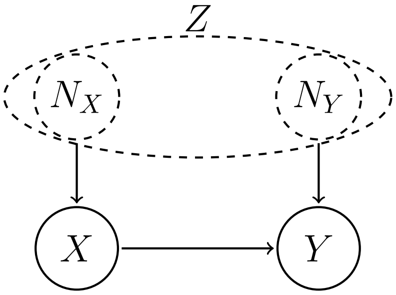


Figure A.1: An example where $\{X, Y\}$ is a causally sufficient set of variables in the sense of Definition 7 but not in the sense of the usual causal sufficiency definition found in the literature, since $Z := (N_X, N_Y)$ is a direct cause of both X and Y w.r.t. $\{X, Y, Z\}$.