Causality: fundamental principles and tools

Irene Balelli, Safaa Al-Ali, Elise Dumas, Judith Abécassis

Chapter 14

Causality: fundamental principles and tools

Irene Balelli[a], Safaa Al-Ali[a], Elise Dumas[b], and Judith Abecassis[c]

[a]Centre Inria d'Université Côte d'Azur, Epione Team, Valbonne, France, [b]EPFL (École polytechnique fédérale de Lausanne), Lausanne, Switzerland, [c]Centre Inria de Saclay, CEA, Soda Team, Palaiseau, France

ABSTRACT

The goal of this chapter is to provide a gentle introduction to Causal Learning (CL), and motivation for its application to medical image analysis, seeking for more robustness against data and domain drifts, and a reliable tool to answer conterfactuals questions and get improved interpretability. The probabilistic formalism at the basis of CL will be introduced, along with basic definitions and assumptions. A number of classical methods to perform causal data analysis (both to establish the causal data generating structure, and to intervene on it) will be illustrated, using simple synthetic datasets. Scaling up to high dimensional and complex data such as medical images is not trivial, and requires the combination of classical CL and modern Deep/Machine Learning techniques: this topic will be further developed in Chapter 17.

KEYWORDS

Causal learning, Discovery, Inference, Bias, Counfounder, Counterfactual queries

14.1 INTRODUCTION

Over the past few decades, the number of data-driven machine learning (ML) and deep learning (DL) methods designed to solve a variety of tasks in medical image analysis, such as segmentation, detection, classification or diagnosis, has exploded. Their accuracy and prediction ability have shown a tremendous improvement, while dealing with a variety of image modalities (*e.g.* CT, MRI, PET, Ultrasound), with applications spanning from oncology to cardiology, neurology and many others [1]. Daily clinical practice is increasingly benefiting from such powerful computational tools, some of which have reached super-human levels of performance.

However, despite their success and utility, purely data-driven approaches are known to suffer from limited robustness and generalizability when confronted with domain shift (or data drift) [2, 3], *i.e.* a shift from the training dataset to the target real-world dataset, which may come from a population shift (*e.g.*

due to different data acquisition protocols and/or devices) or from a distribution shift (*e.g.* an under-represented sub-population of interest in the training set, for instance with respect to a specific targeted disease, or even the emergence of new diseases). In addition, the real-world deployability of ML/DL models by the healthcare end-users (clinicians, physicians, or patients) is tightly related to the users' ability to understand and trust the algorithmic prescriptions, the encoded assumptions and, from a higher level perspective, the path leading from the initial inputs to the model's outputs, whereas ML and DL approaches mostly operate as black boxes preventing a smooth machine-user interaction.

To alleviate both the generalizability and the transparency issues, causal learning (CL) has emerged as a promising candidate solution, and is attracting growing interest in the healthcare community. The seminal work of Turing awarded J. Pearl [4, 5] posed the mathematical foundations of causality and causal reasoning, and established a clear formalism to represent the data-generating process and infer the causal effects of interventions on its variables. Rubin and Neyman [6, 7] also strongly contributed to the development of this field, by introducing an alternative statistical framework for causal inference.

The aim of this Chapter is to present the fundamentals of causality and its applicability to healthcare. In Sec. 15.2 standard notations and basic definitions of causal reasoning are recalled and illustrated. Sec. 15.3 is dedicated to causal discovery, the branch of causality that attempts to disentangle the causal relationships between the retained variables, hence learn the data generating process. Sec. 15.4 focuses on causal inference, *i.e.* on estimating the causal effects of external interventions on the system, after its data generating process has been established. For the sake of completeness, Pearl's formalism will be favored in 15.3, while 15.4 will rely mainly on Rubin's framework.

All code examples for this Chapter are available in this GitLab repository, which includes several notebooks for illustration of the presented tools. The `README` file provides all instructions for the requirements.

14.2 THE BASIS FOR BUILDING A CAUSAL REASONING

A common sense definition of *causality* can be stated as follows: if X causes Y, then an intervention in the value or state of X implies a change in the value or state of Y, whereas the reverse is not expected. Consider, for example, the relationship between two *variables*: age and cognitive decline. It is well known that aging induces cognitive decline, but abnormal cognitive decline won't increase aging speed. Hence, aging is a *cause* of cognitive decline, the *effect*. It is intuitive to represent *causal knowledge* using nodes for variables of interest (here age and cognitive level), and arrows pointing from each cause to its effects: this forms a *directed graph* (Fig. 15.1 **(a)**). Let us now include an additional variable, formal education, a protective factor against cognitive decline: a new arrow will

(a) (b)

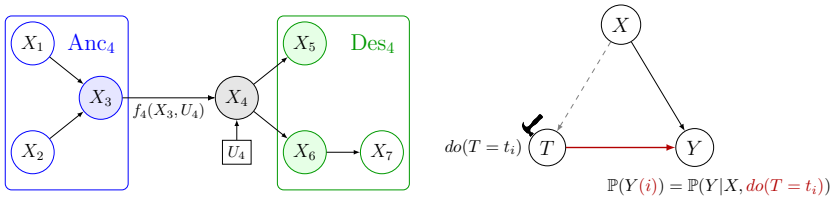
FIGURE 14.1 An illustrating example. **(a)** Two variables, X_1, the age, and X_2, the cognitive level, are considered at first: the directed arrows stress the causal directions. **(b)** A new variable is added, X_3, the education level. This diagram can be iteratively populated with additional variables.

point from education to cognitive decline (Fig. 15.1 **(b)**). Based on the domain knowledge from Fig. 15.1 **(b)**, given a dataset containing age, cognitive level and former education level, one would expect to find a statistical correlation between age and cognitive level, and between formal education and cognitive level. Further, a correlation between age and education is also expected within strata of cognitive levels, despite they are not causally linked. However, statistical correlation is not informative about causation *per se*, making the knowledge of the causal paths between variables a necessary condition for a robust interpretation and analysis of collected data. Nowadays in healthcare, a huge variety of data can be generated on each patient, such as genetic information, biomarkers, the clinical history, up to imaging features. The underlying causal relationships between all these variables may not be already fully established: is it possible to recover it from available data? This is the central question tackled by causal discovery (Sec. 15.3). In order to get there, a graphical causal formalism, first proposed by J. Pearl, will be introduced.

Let $\mathbf{X} := \{X_1, \ldots, X_N\}$ be a set of N *endogenous* variables (*i.e.* variables one wants to include by design in the model), which are assumed being ordered in a cause-effect manner (Fig. 15.2 **(a)**). For each $X_i \in \mathbf{X}$, let PA_i denotes the parents of X_i, *i.e.* the set of all variables in $\mathbf{X}$ which directly causes X_i: $\mathrm{PA}_i := \{X_j \in \mathbf{X} \mid X_j \rightarrow X_i\}$. Similarly, the ancestors of X_i, Anc_i, contains all nodes preceding X_i in a causal cascade, and the children and descendants of X_i, Ch_i and Des_i respectively, define the sets of nodes which causally follow (directly and indirectly) X_i. Formally:

- $\mathrm{Anc}_i := \{X_j \in \mathbf{X} \mid \exists \text{ causal path from } X_j \text{ to } X_i\}$,
- $\mathrm{Ch}_i := \{X_j \in \mathbf{X} \mid X_i \rightarrow X_j\}$,
- $\mathrm{Des}_i := \{X_j \in \mathbf{X} \mid \exists \text{ causal path from } X_i \text{ to } X_j\}$,

where a path from a node X_j to a node X_i is a set of consecutive edges from X_j to X_i following non-intersecting nodes. A path is said to be causal if it follows the direction of the arrows. Clearly: $\mathrm{PA}_i \subset \mathrm{Anc}_i$, $\mathrm{Ch}_i \subset \mathrm{Des}_i$. Moreover, knowing PA_i, X_i is independent from all his previous ancestors ($\mathrm{Anc}_i/\mathrm{PA}_i := \{X_i \in \mathrm{Anc}_i \text{ and } X_i \notin \mathrm{PA}_i\}$), in the causal order sense, meaning that if all variables in PA_i are observed, knowledge about previous ancestors of X_i won't add any extra information to predict its current state. To each variable $X_i \in \mathbf{X}$, an *exogenous*



(a) Elements of a BCN

(b) Randomization or causal inference

FIGURE 14.2 Bayesian Causal Networks and interventions. **(a)** Let us consider a set of 7 endogenous variables X_i (represented with circles) and focus on X_4. The exogenous noise variable corresponding to X_4, U_4, is represented with a rectangular vertex. The ancestors of X_4 are highlighted in the blue bounded region (the parent node of X_4 is filled in blue), while its descendants are contained in the green bounded region (children nodes are filled in green). **(b)** Let X be a patient characteristic (or covariate), *e.g.* age, which both affects the outcome of interest Y *e.g.* the pace of a cancer progression, and the choice of a treatment T, *e.g.* chemotherapy, immunotherapy or a combination. In a randomized clinical trial the randomization makes the treatment assignment for each participant independent from his/her baseline characteristics: this can be investigated through the do operator. An external intervention over the treatment assignment, $do(T = t_i)$ entails a modification of the joint distribution of (X, T, Y) as it was initially defined. The modified probability distribution results from the removal of all incoming causal links on the intervened variable, and the substitution of its corresponding functional with the prescription $T = t_i$.

unobserved noise variable $U_i \in \mathbf{U} := \{U_1, \ldots, U_N\}$ is associated, to model the unexplained variability of X_i, *i.e.* the variability of X_i which is not related to the variation of any other endogenous variable $X_j, j \neq i$. All $U_i \in \mathbf{U}$ are supposed to be mutually independent, which implies $U_i \perp\!\!\!\perp U_j, \forall j \neq i$. Consequently, the joint distributions of $\mathbf{X}$ and $\mathbf{U}$ are respectively:

$$\mathbb{P}(\mathbf{X}) = \prod_{i=1}^{N} \mathbb{P}(X_i|\text{PA}_i) \quad (14.1) \qquad \mathbb{P}(\mathbf{U}) = \prod_{i=1}^{N} \mathbb{P}(U_i) \quad (14.2)$$

Eq. (15.1) reflecting the Markovian density of $\mathbf{X}$ is called the *independent causal mechanisms principle*. Finally, the last ingredient needed to define a *structural causal model* (SCM) is a set of functionals $\mathbf{F} := \{f_1, \ldots, f_N\}$ relating in a deterministic way X_i, PA_i and U_i, for all $i = 1, \ldots, N$:

$$X_i = f_i(\text{PA}_i, U_i). \quad (14.3)$$

A very convenient way to graphically illustrate all the components introduced so far is by using a directed *acyclic* graph (DAG) - $\mathcal{G}$, where *directed* means that all arrows point in exactly one direction, while *acyclic* indicates that there exists no directed causal path from any two variables (X_i, X_j) in $\mathcal{G}$ so that $X_i = X_j$, or equivalently $\forall i = 1, \ldots, N, \text{Anc}_i \cap \text{Des}_i = \emptyset$. A *Bayesian (causal) network* (BCN

- Fig. 15.2 **(a)**) is finally obtained when a SCM is coupled with its corresponding DAG. A large amount of information can be encoded in a BCN, which serves to build a clear causal reasoning, and select the variables to be controlled for to reach conditional independence, and properly estimate the causal effect of interest. Some node configurations are of particular relevance and deserve to be defined here. Fig. 15.2 **(a)** will be used as a reference to illustrate them. In the following, the conditional independence of two nodes X_i, X_j given X_k is denoted with the symbol $\perp\!\!\!\perp_{X_k}$.

Root and sink: variables with an empty parents' set or with an empty children' set, respectively. Root (resp. sink) variables are independent from all other variables in $\mathbf{X}$ except their descendants (resp. except parents, given their parents): $X_{\text{root}} \perp\!\!\!\perp \mathbf{X}/\text{Des}_{\text{root}}$ and $X_{\text{sink}} \perp\!\!\!\perp_{\text{PA}_{\text{sink}}} \mathbf{X}/\text{PA}_{\text{sink}}$. In Fig. 15.2 **(a)**, X_1 and X_2 are root variables, while X_5 and X_7 are sink variables.

Collider: a common effect of two (or more) variables, such as X_3, a collider for X_1 and X_2. When it happens, as in Fig. 15.2 **(a)**, that X_1 and X_2 are not causally linked, than despite being marginally independent, they become conditionally dependent given their collider: $X_1 \perp\!\!\!\perp X_2$, but $X_1 \not\perp\!\!\!\perp_{X_3} X_2$.

Confounder: a common cause of two (or more) variables, such as X_4 for variables X_5 and X_6. In this case, X_5 and X_6 are marginally dependent, but become conditionally independent given X_4: $X_5 \not\perp\!\!\!\perp X_6$, but $X_5 \perp\!\!\!\perp_{X_4} X_6$.

Mediator: a variable that mediates between a cause and an effect, such as X_6 which mediates the effect of X_4 over X_7. Here, the cause X_4, and the effect X_7, are marginally dependent, but become conditionally independent given the mediator, X_6: $X_7 \not\perp\!\!\!\perp X_4$, but $X_7 \perp\!\!\!\perp_{X_6} X_4$.

The reading and interpretation of a BCN, which provides a clear causal representation of the variables of interest, and their dependencies, have now been clarified. In healthcare, it is crucial to be able to assess the effect of an intervention, such as the administration of a treatment or surgery, on some outcomes, such as disease relapse or mortality. Observational data derived from actual clinical practices are prone to *confounding*, *i.e.* the presence of variables, such as patients' baseline characteristics or their medical history, which affect both the medical intervention decision and its outcomes, and may prevent a reliable analysis of the intervention-outcome causal associations. A gold standard solution to overcome this problem is provided by randomized clinical trials, where the randomization allows to break the links between the (known) confounders and the treatment. However, on the one hand, it is not always possible to carry out clinical trials due, for example, to ethical or financial barriers and, on the other hand, a large amount of observational data on treatment effects is collected on a daily basis: this constitutes a precious source of information, which in addition is not biased by the strict selection process of patients in an experimental setting. Can actionable decisions be made on possible interventions from observational data? This can be investigated through causal inference.

Two main frameworks have been developed for causal inference. The first

one, introduced as well by J. Pearl, is based on a new operator, called the *do* operator, which defines firstly a way to intervene on a BCN, and secondly, a proper formalism to capture the entailed perturbations in the data generating process from a fully probabilistic perspective (Fig. 15.2 **(b)**). This framework will not be covered further in this chapter. Alternatively, Rubin [6] and Imbens and Rubin [7] formalized a fully statistical approach called the *potential outcome* (PO) framework, widely used to reason about the effect of a treatment, in particular when disposing of non-randomized (or observational) data, where the treatment class assignment can not be controlled. Rubin's PO framework will be further developed in Sec. 15.5. Of note, theoretical bridges exist between SCMs and Pearl's *do* operator and the PO framework [8, 9].

In the following sections, some methods and existing tools will be illustrated, for causal discovery (Sec. 15.3), to establish the most likely underlying BCN relating the features at hand, and for causal inference (Sec. 15.4), to intervene over a given BCN, and then quantify the downstream effects of the intervention across the graph.

14.3 CAUSAL DISCOVERY OR THE QUERY FOR THE DATA GENERATING PROCESS

In healthcare, it is common to deal with complex systems, where several variables of interest can interact with each other and contribute to the evolution of the underlying process. Domain experts may not have complete knowledge of these relationships, which can be partially or completely unknown: this motivates the use of causal discovery. Causal discovery aims to uncover cause-and-effect relationships between the variables under consideration, based on a set of observations, thereby improving understanding and insight into the studied condition. Several causal search algorithms have been developed over the last decades to unveil causal connections between variables. Existing methods differ both in the assumptions they rely on and in the type of data they use as input, which can be observational data or a mix of experimental and observational data.

Firstly, some essential definitions and assumptions used during the stage of discovery of the causal graph will be introduced. Secondly, several causal discovery algorithms will be discussed and illustrated with practical examples.

14.3.1 Definitions and assumptions

Let $\mathcal{G}_{\mathbf{X}}$ denote a causal graph on the node set $\mathbf{X}$.

Skeleton: the fully undirected graph associated to $\mathcal{G}_{\mathbf{X}}$.

Causal sufficiency (CS): $\mathcal{G}_{\mathbf{X}}$ satisfies the CS if for every pair $X_i, X_j \in \mathbf{X}$, all their common causes are assumed to be observed and modeled in $\mathcal{G}_{\mathbf{X}}$.

Blocking path: X_{bl} is said to be blocking a path between X_i and X_j if (i) X_{bl} is a mediator between X_i and X_j, or (ii) X_{bl} if a confounder for X_i and X_j, or (iii) X_{bl} is not a collider in the path from X_i to X_j, nor a descendant of a

collider.

$d-$**separation:** given a set of pairwise disjoint subsets $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3 \subset \mathbf{X}$, $\mathbf{V}_3$ $d-$separates $\mathbf{V}_1$ and $\mathbf{V}_2$ if it blocks all the paths between nodes in $\mathbf{V}_1$ and nodes in $\mathbf{V}_2$. This will be denoted: $\mathbf{V}_1 \perp_{\mathbf{V}_3} \mathbf{V}_2$.

$v-$**structure:** a triplet (X_1, X_2, X_3) so that X_1, X_2, X_3 are linked in the skeleton, $X_1 - X_2 - X_3$, and X_2 is a collider or a confounder.

Causal Markov condition (CMC): $\mathcal{G}_{\mathbf{X}}$ satisfies the CMC if $\forall X_i \in \mathcal{G}_{\mathbf{X}}$, $X_i \perp\!\!\!\perp_{\mathrm{PA}_i} \mathbf{X}/(\mathrm{Des}_i \cup \mathrm{PA}_i)$. In DAGs, the CMC is equivalent to the factorization in Eq. (15.1).

Global Markov condition (GMC): The GMC holds if for every pairwise disjoint subsets $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3 \subset \mathbf{X}$, $\mathbf{V}_1 \perp_{\mathbf{V}_3} \mathbf{V}_2 \Rightarrow \mathbf{V}_1 \perp\!\!\!\perp_{\mathbf{V}_3} \mathbf{V}_2$.

Faithfulness: Faithfulness reverses the GMC: $\forall\, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3 \subset \mathbf{X}$ pairwise disjoint, $\mathbf{V}_1 \perp\!\!\!\perp_{\mathbf{V}_3} \mathbf{V}_2 \Rightarrow \mathbf{V}_1 \perp_{\mathbf{V}_3} \mathbf{V}_2$.

14.3.2 Causal discovery for cross-sectional data

Cross-sectional (or stationary) data refers to a set of observations collected at a specific time point, each observation belonging to a different individual: causal discovery applied to such data is a very active research field. Two main families of causal discovery algorithms can be distinguished: constraint-based and (Bayesian) score-based algorithms [10, 11].

Constraint-based methods rely on the CMC and the faithfulness assumption. They are based on the search for conditional independence relationships between the observed variables through some appropriate statistical tests of conditional independence, and return the graph(s) consistent with such constraints. They can be applied to a wide range of data types, including continuous, categorical, and textual data, and have the advantage of not making any assumption on the structure of the functionals f_i, and the distribution of the variables $X_i \in \mathbf{X}$. Nevertheless, a major limitation of such methods is due to the curse of dimensionality, since the number of possible conditional independencies to be tested grows exponentially with the number of variables.

On the other hand, score-based methods perform a search for all possible causal graphs over $\mathbf{X}$, trying to find the model that best fits the data. This is done by maximizing a score typically derived from the likelihood of the data given the graph $\mathcal{G}_{\mathbf{X}}$ (e.g. Bayesian Information Criterion *BIC*, among others), according to the factorization imposed by the graph $\mathcal{G}_{\mathbf{X}}$ through the CMC. Such assumed factorization allows efficient search-and-score learning, reducing the number of computations needed for scoring each change on the graph $\mathcal{G}_{\mathbf{X}}$. Nevertheless, score-based methods require careful consideration of the priors and the scoring function with respect to the domain-specific context in which they are applied. Of note: some hybrid methods have also been developed [12], but won't be detailed in this chapter.

In the following, classical constraint- and score-based causal discovery algorithms will be introduced. They will be illustrated through practical examples

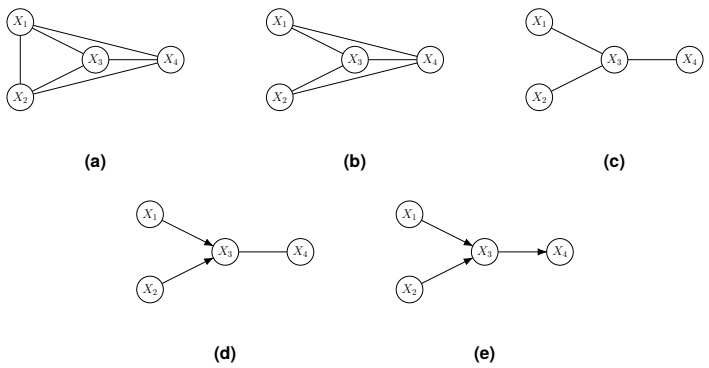


FIGURE 14.3 PC algorithm steps over $\mathbf{X} := \{X_1, X_2, X_3, X_4\}$ from Fig. 15.2 **(a)**. **(a-c)** skeleton discovery phase, **(d-e)** edge direction phase.

on synthetic data generated according to the graph in Fig. 15.2 **(a)**, and assuming linear relationships. The code presented in this section relies on the python package `causallearn` [13]. Other packages for causal discovery exist, both in Python and R (see [10, 11]).

Data generation can be easily performed using Python: a code is available in the GitLab repository.

Peter and Clark's (PC) [14] is one of the first causal discovery algorithms developed, and is still widely used in the community. It is a constraint-based method and consists of two steps [15, 11]:

1. *Skeleton discovery*: the algorithm starts with a complete undirected graph over all the observed variables (Fig. 15.3 **(a)**). For every pair of adjacent variables, PC first tests their independence: if it holds, the edge connecting them is removed (Fig. 15.3 **(b)**). Then, conditional independence with respect to subsets of variables of increasing size is iteratively tested, and the skeleton is updated accordingly (Fig. 15.3 **(c)**).
2. *Edge orientation:* PC searches for v-structures, and uses the conditional independence test to orient their edges (Fig. 15.3 **(d)**). Finally, it performs the orientation propagation to the remaining undirected edges (Fig. 15.3 **(e)**): for every triplet (X_1, X_2, X_3) such that $X_1 \rightarrow X_2 - X_3$, if X_1 and X_3 are not adjacent in $\mathcal{G}_{\mathbf{X}}$, it concludes $X_2 \rightarrow X_3$.

PC can be imported from `causallearn`, which enables to set some customized advanced parameters, such as the conditional independence test (Fisher's Z by default). The output graph for PC with the default parameters is shown in Fig. 15.4 **(a)**.

```python
# import PC method from causallearn package
from causallearn.search.ConstraintBased.PC import pc

# Apply PC to numpy array data=(n_subjects,n_variables)
cg= pc(data=data)

# Use PC with kernel-based conditional independence test
from causallearn.utils.cit import kci
cg= pc(data=data, indep_test=kci, kernelZ='Gaussian')

# Create and visualize the causal graph
from causallearn.utils.GraphUtils import GraphUtils
cg.draw_pydot_graph()
pyd = GraphUtils.to_pydot(cg.G)
```

Of note, PC may fail in orienting some edges, so that the final output graph may be only partially directed. Moreover, PC is order sensitive, meaning that changing the order in which variables are considered during the skeleton discovery phase may affect the final output skeleton, hence the subsequent edge orientation phase. An alternative version, **PC-stable** [16], was later proposed to address this issue. PC-stable stores the neighbors (or adjacency) set of every node at each step of the skeleton discovery phase, so that an edge deletion will not affect the conditional dependence tests of other variable pairs in the current stage. Other PC variants have also been developed, in particular, **conservative PC** [17], more cautious in the edge orientation phase.

A limitation of PC and its variants is that they all assume causal sufficiency, a quite strong assumption. The **Fast Causal Inference** (FCI) [14] method allows to relax CS by assuming that latent confounders may exist. It starts with a skeleton discovery phase similar to PC, but then uses $d-$separation for the edge orientation phase. Several types of relationships can be displayed in the FCI's output graphs, among which bi-direction, which denotes the presence of an unobserved confounder between the linked variables. Fig. 15.4 **(b)** shows an example of an output graph with FCI applied to the synthetic data generated from Fig. 15.2 **(a)**. Similarly to PC, several variants of FCI have been proposed, mostly to speed up the algorithm, such as Anytime FCI [18] and Really FCI [19].

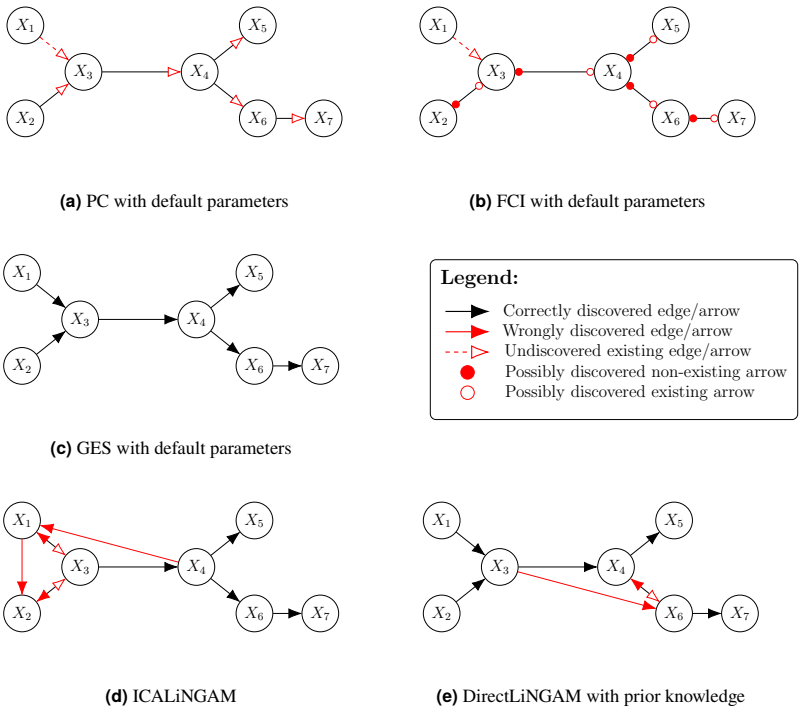
Among score-based methods, the **Greedy Equivalence Search** (GES) [20], together with its variants is widely applied. Unlike PC and FCI, GES starts from an empty graph. It consists of two phases: a forward equivalence search, where edges are iteratively added, followed by a backward iterative search, where edges are iteratively removed. In both phases, GES evaluates every new resulting graph, using a scoring function to assess the trade-off between the model complexity and the quality of data fit. To avoid redundancy during the greedy search, GES looks for equivalence classes among the graph structures.

The default score for GES is BIC (see the corresponding output graph in Fig. 15.4

(c)), but other are available in `causallearn`, such as negative k-fold cross-validated log-likelihood.

```
# import GES method from causallearn
from causallearn.search.ScoreBased.GES import ges

# Apply GES to df, and recover the graph (['G'])
cg = ges(data)['G']

# Use k-fold cross-validated log likelihood score
cg = ges(data,score_func='local_score_CV_general')['G']
```

More recently, causal discovery algorithms based on functional causal models



(a) PC with default parameters

(b) FCI with default parameters

(c) GES with default parameters

(d) ICALiNGAM

(e) DirectLiNGAM with prior knowledge

FIGURE 14.4 DAGs obtained from different causal discovery algorithms applied to synthetic data generated from the causal graph in Fig. 15.2 **(a)**. **(a)** PC with Fisher's Z test function (the default in `causallearn`): most edges are correctly identified, but PC fails to unveil all directions. **(b)** Like PC, FCI identifies the majority of links but fail in the edge orientation phase: of note, FCI clearly highlights its doubts in edge direction using the dot arrowhead. **(c)** The score-based method GES with BIC score identifies correctly the ground truth graph. **(d)** ICA-LiNGAM shows some erroneous dependencies between X_1, X_2 and X_3: this can be corrected by **(e)** imposing that X_1 and X_2 are root.

were proposed, such as the Linear non-Gaussian acyclic model **LiNGAM** [21], which assumes linear relationships between each node and its parents, *i.e.* $\forall X_i \in \mathbf{X}, X_i = \sum_{X_j \in \mathrm{PA}_i} b_{ji} X_j + U_i$, where U_i is a centered non-Gaussian random variable with non-zero variance. LiNGAM assumes faithfulness and the acyclicity of $\mathcal{G}_{\mathbf{X}}$. It aims to estimate the matrix $B := b_{ij}$, knowing that it can be permuted to a strictly lower-triangular matrix, due to acyclicity. Several extensions of LiNGAM have been proposed, such as **ICA-LiNGAM**, which assumes that the observed variables' dependence may be due to unobserved latent confounders and performs an Independent Component Analysis (ICA) before applying LiNGAM, or **DirectLiNGAM** [22], which improves the robustness of LiNGAM estimation method while providing formal convergence guarantees.

```python
# Import lingam-based methods from causallearn
from causallearn.search.FCMBased import lingam

# Fit ICA-LiNGAM on data
model = lingam.ICALiNGAM(random_state=42)
model.fit(data)

# Make the causal graph using the adjacency matrix
def make_graph(adjacency_matrix, **kwargs):
    # ...

G = make_graph(model.adjacency_matrix_)
```

The output graph is shown in Fig. 15.4 **(d)**.

In some contexts, the causal graph may already be partially established through expert knowledge, hence some causal links do not need to be discovered. It is of interest to incorporate this prior knowledge into the system in order to drive the causal discovery search toward the true overall causal graph.

It may be known that X_1, X_2 are root. This information can be encoded in a matrix `M`:= m_{ji}, an adjacency matrix with partial information, *i.e.* so that $m_{ji} = 1$ (resp. 0) for all known existing (resp. absent) causal links $X_i \to X_j$, and $m_{ji} = -1$ otherwise, *i.e.* when no prior knowledge is available. Fig. 15.4 **(e)** shows the output graph of DirectLiNGAM after injecting the available prior knowledge through matrix M, so that $\forall j = 1, \ldots, 7, m_{1i} = m_{2i} = 0$ ($m_{ji} = -1$ for $j \notin \{1, 2\}$).

```python
# Define a prior knowledge matrix
def make_prior(n_vars, root, sink, *args):
    # ...
M = make_prior(n_vars=7, root=[0,1])

# Fit DirectLiNGAM on data using M as prior knowledge
model = lingam.DirectLiNGAM(prior_knowledge=M)
model.fit(data)
```

14.3.3 Causal discovery for time series data

Up to now, only stationary data have been considered, *i.e.* data which are representative of a specific time stamp. Nevertheless, in many situations it may be interesting to analyze the temporal evolution of a medical condition, *e.g.* a disease, in case observations collected over a period of time are made available: these are called time series or longitudinal data. The causal discovery approaches presented so far are not adapted to account for temporal relationships in their causal search: the temporal dimension has to be explicitly included in the model.

Most classical causal discovery algorithms already have their own extension for time series data. For instance, the **PCMCI** method [23] is derived from PC. As PC, it is a constraint-based method: it assumes time-lagged dependencies and uses the momentary conditional independence (MCI) test. The **time series FCI** (tsFCI) [24] is an adaptation of FCI which considers the temporal order of observations and seeks to identify causal links stable both within and between time points. Among LiNGAM-based methods, **VarLiNGAM** [25] is well adapted to deal with time series data and combines autoregressive and non-Gaussian models to estimate instantaneous and lagged causal effects.

Probably one of the most known frameworks for causal discovery with time series data is Granger-causality (GC), introduced by Granger in 1969 [26]. GC-based methods assume no latent confounders and no instantaneous dependencies, *i.e.* only past values of the variables can be used to predict the current status of the system. Following GC, **Generalized Vector AutoRegression** (GVAR) [27] has been recently developed and can be applied to multivariate time series under nonlinear dynamics. Another interesting approach is **Amortized Causal Discovery** (ACD) [28], a GC-based method which can combine samples coming from distinct causal graphs but that share common dynamics. It consists of two blocks: an encoder to learn the causal graph and a decoder to simulate the dynamics of the system for the next time-step. For a more comprehensive survey of causal discovery methods for time series data you may refer *e.g.* to [29].

14.4 CAUSAL INFERENCE AND COUNTERFACTUAL QUESTIONS

Once a causal graph has been obtained, either through expert knowledge or using a causal discovery algorithm, it can be used to perform causal inference, that is, quantify the effect of an intervention on one variable of the graph, denoted T for treatment, on another variable of the graph, denoted Y, the outcome.

The definition of the main causal estimands introduced throughout Sec. 15.4, *i.e.* the quantities of interest to be estimated, will rely on the *Potential Outcomes* framework, formalized by Neyman and Rubin [6, 7]. The variable T will be referred to as the intervention or the treatment indifferently. Units that do not receive the intervention are sometimes called control units. The methods presented in this section are designed for observational data, where the researchers do not control the treatment assignment mechanism, though very similar approaches

can be applied to data where the treatment was randomized. Let us consider a set of n observed units, with a binary treatment: $T_i = 1$ if a unit $i \in \{1, \ldots, n\}$ is treated, $T_i = 0$ otherwise. Let Y_i be the observed outcome of unit i and $Y_i(t)$ the *potential* outcome for this unit, with t in $\{0, 1\}$ the intervention: this is the outcome that would have occurred if the intervention had been t. In the data, only one of those potential outcomes is observed, but the notion of potential outcome allows to reason about what the outcome would have been under a different intervention value. The unobserved potential outcome is sometimes called the *counterfactual* outcome since it is "contrary to" the facts. The contrast between two potential outcomes defines the *individual treatment effect* (ITE):

$$ITE_i := Y_i(1) - Y_i(0). \tag{14.4}$$

The ITE can never be observed, as a unit either receives the intervention or not: this is the fundamental problem of causal inference [30]. However, this difficulty can be circumvented by reasoning at the scale of the population and consider the *average treatment effect*:

$$ATE := \tau := \mathbb{E}[Y(1) - Y(0)]. \tag{14.5}$$

Finally, if one is interested in the heterogeneity of the treatment effect depending on the unit's characteristics, the *conditional average treatment effect* can be considered:

$$CATE := \tau(\mathbf{x}) := \mathbb{E}[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}], \tag{14.6}$$

where $\mathbf{X}$ represents covariates of interest.

For a given causal estimand, inference consists of two stages: first, the *identification*, to verify that the available data are sufficient to eliminate all potential sources of bias, and second, the *estimation*, to actually calculate the causal effect using the data. The average treatment effect (ATE) will be privileged in this section, but similar concepts and techniques can be applied to other estimands. The code presented in this section relies on the `DoWhy` Python package [31, 32], which covers all the steps of a causal analysis. `DoWhy` also includes the `EconML` functionalities, which offer highly flexible estimators based on machine learning (ML) models. There are numerous alternatives for R practitioners [33].

14.4.1 Identification of the average causal effect

The causal estimand of interest is usually expressed as a function of potential outcomes under two alternative treatments. However, both potential outcomes are never observed, so the identification phase consists of moving from a causal estimand to a statistical estimand that depends only on observable random variables. For the ATE, this transformation requires three assumptions [34]:

Consistency (or SUTVA Stable Unit Treatment Value Assumption): $Y = (1-T)Y(0) + TY(1)$. Consistency establishes the link between the observed outcome and the potential outcomes through the actual intervention. It is achieved when there is no interference between units, and only one version of the intervention.

Ignorability (or Unconfoundedness): $\{Y(0), Y(1)\} \perp\!\!\!\perp_{\mathbf{X}} T$. Ignorability states that the potential outcomes and the treatment assignment mechanism are conditionally independent given the covariates, meaning that there is no unmeasured confounder. This is a very strong and untestable assumption.

Positivity (or Overlap): $\exists\, \eta > 0, \forall \mathbf{x} \in \mathcal{X}, \eta < \mathbb{P}(T = 1|\mathbf{X} = \mathbf{x}) < 1 - \eta$ (with $\mathcal{X}$ the support of $\mathbf{X}$). Positivity means that the intervention assignment is not deterministic, so every unit has a chance to receive the intervention or not. This is necessary to compare the potential outcomes under intervention or no intervention for all possible covariates combinations.

Under those assumptions, the ATE is identified:

$$\begin{aligned} ATE &= \mathbb{E}[Y(1) - Y(0)] \\ &= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(1) - Y(0)|\mathbf{X}]] \\ &= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(1)|\mathbf{X}, T = 1]] - \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(0)|\mathbf{X}, T = 0]] \quad \textit{(ignorability and positivity)} \\ &= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y|\mathbf{X}, T = 1]] - \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y|\mathbf{X}, T = 0]] \quad \textit{(consistency)}. \end{aligned} \tag{14.7}$$

Indeed, Eq. (15.7) shows that the ATE can be expressed only with observable random variables ($Y, \mathbf{X}$ and T). In practice, **consistency** is established by considering the data collection mechanism. Regarding the **positivity** assumption, it can be tested from the data [35]. Finally, the **ignorability** assumption can be assessed using the properties of the causal graph: the variables in $\mathbf{X}$ constitute a sufficient adjustment set if $\mathbf{X}$ d-separates T and Y. If the treatment is randomly assigned, no adjustment is needed to enforce the ignorability assumption.

Some methods to estimate the ATE from the observed data will now be presented. For illustration purposes, practical examples will be proposed using synthetic data generated with `DoWhy` from the causal graph in Fig. 15.5.

```
import numpy as np
import dowhy, dowhy.datasets

# Set a seed for reproducibility
np.random.seed(18)
# Create a synthetic dataset. Note:
data = dowhy.datasets.linear_dataset(
    num_common_causes=5,
    num_instruments = 1,
    num_effect_modifiers=1,
    treatment_is_binary=True,
    num_samples=3000,
    num_discrete_common_causes=1,
```

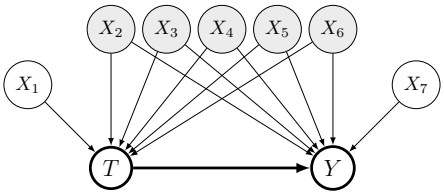


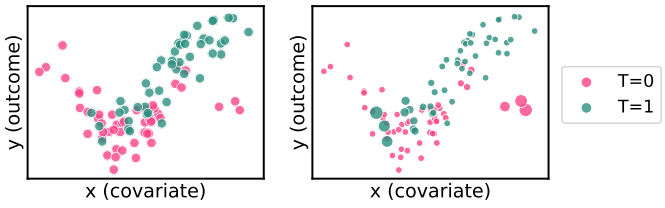
FIGURE 14.5 Causal graph underlying the example for estimation approaches. The ATE is identified, and unconfoundedness is achieved by adjusting on variables $X2$ to $X6$ (the grey filled nodes). X_1 and X_7 affect only either the treatment or the outcome and do not need to be controlled for.

```
    beta=10,stddev_treatment_noise=10)

# Create a CausalModel object
model = dowhy.CausalModel(data=data['df'],
                          treatment=data['treatment_name'],
                          outcome=data['outcome_name'],
                          graph=data["gml_graph"])

# Identify the causal effect
identified_estimand = model.identify_effect(
    optimize_backdoor=True,
    proceed_when_unidentifiable=True)

# True ATE
print(np.round(data['ate'], 2))
# 9.77
```

14.4.2 Estimation of the average causal effect

The objective of this step is to construct an estimator with good statistical properties. The two main families of estimators will be presented: they are based on either estimating the probability of receiving the treatment or on modeling of the outcome by regression. More recent methods that combine both traditional approaches into more robust estimators, will also be introduced.

A first idea to adjust for confounding variables is to reweight each observed sample by the **propensity score**, which is the probability of receiving the intervention given the covariates: $e(\mathbf{X}) = \mathbb{P}(T = 1|\mathbf{X})$. The first step of the estimation is to model this quantity using logistic regression. The estimated scores are then plugged in the inverse-propensity weighting (IPW) estimator for the ATE:



(a) Original data **(b)** Reweighted data

FIGURE 14.6 Illustration of the intuition behind the IPW estimator. The outcome of the treated and the control individuals can be compared to estimate the intervention effect after reweighting to create subpopulations for which the intervention assignment does not depend on the unit's characteristics. **(a)** Original data. **(b)** Reweighted observations: units that received the intervention but had a low probability of receiving it are given more importance (and conversely for units that did not receive the intervention).

$$\widehat{ATE}_{\text{IPW}} := \frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_i Y_i}{\hat{e}(\mathbf{X}_i)} - \frac{(1-T_i)Y_i}{1-\hat{e}(\mathbf{X}_i)}\right) \tag{14.8}$$

The intuition behind $\widehat{ATE}_{\text{IPW}}$ is that the weights balance the distribution of the covariates between the intervention and the control groups, as shown in Fig. 15.6. The reweighted observations can be handled as if the intervention was assigned at random. If the propensity score model is consistent, and the three identifiability assumptions hold, the IPW estimator is consistent for the ATE.

```
estimate_ipw = model.estimate_effect(identified_estimand,
    method_name="backdoor.propensity_score_weighting",
    target_units = "ate",
    method_params={"weighting_scheme":"ips_stabilized_weight"
    },
    confidence_intervals="bootstrap")

print("ATE Estimate: {:.2f}, confidence interval: {}.".format
    (np.round(estimate_ipw.value,2),np.round(estimate_ipw.
    get_confidence_intervals(),2)))
# ATE Estimate: 10.13, confidence interval: [9.97 10.26].
```

Another approach to account for confounding variables in the estimation is to directly model their impact on the outcome Y by regression. In the case of the g-formula plug-in estimator, also called S-learner estimator (S for single), one regression model $Y = \mu(T, \mathbf{X})$ is fitted on the data, in which the outcome is regressed on both the intervention T and the covariates $\mathbf{X}$. Alternatively, for the T-learner estimator, two models are fitted: $Y = \mu_1(\mathbf{X})$ on the treated observations, and $Y = \mu_0(\mathbf{X})$ on the controls. Here, the outcome is regressed

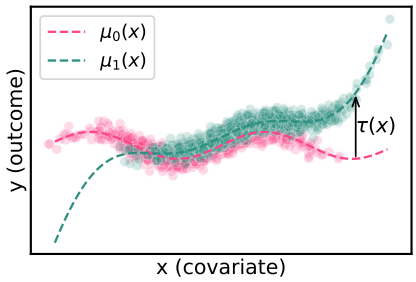


FIGURE 14.7 Illustration of the intuition behind the T-learner. Two regression models are fitted to the data. This approach makes intervention effect heterogeneity explicit, as variations can be seen in the distance between the two response surfaces depending on the covariate value.

only on the covariates. In both cases, the predictions of the model(s) are used to compute the difference in the two alternative potential outcomes:

$$\widehat{ATE}_S = \frac{1}{n}\sum_{i=1}^{n} \hat{\mu}(1, \mathbf{X}_i) - \hat{\mu}(0, \mathbf{X}_i) \tag{14.9}$$

$$\widehat{ATE}_T = \frac{1}{n}\sum_{i=1}^{n} \hat{\mu}_1(\mathbf{X}_i) - \hat{\mu}_0(\mathbf{X}_i) \tag{14.10}$$

The T-learner estimator can handle treatment effect heterogeneity and is also more sensitive to a small magnitude of the effect, as the model is forced to use the treatment variable. However, it requires more data than the S-learner, since the regression models are fitted only on a part of the dataset. Regression-based estimators are consistent only if the regression models are consistent, and the three identifiability assumptions hold. Regression-based approaches are a very natural way to detect and visualize heterogeneous intervention effects, *i.e.* where the effect depends on the covariates, as represented in Fig. 15.7. In this case, it is more informative to consider the CATE estimand, rather than the ATE.

```
from sklearn.linear_model import LinearRegression

# S-Learner
estimate_s = model.estimate_effect(
    identified_estimand=identified_estimand,
    method_name='backdoor.econml.metalearners.SLearner',
    target_units='ate',
    confidence_intervals=True,
    method_params={'init_params': {'overall_model':
    LinearRegression()},
        'fit_params': {'inference': 'bootstrap'}})

ci_s = [c.mean() for c in estimate_s.get_confidence_intervals
```

```
    ()]

print("ATE Estimate: {:.2f}, confidence interval: {}.".format
    (np.round(estimate_s.value,2),
    np.round(ci_s,2)))
# ATE Estimate: 9.77, confidence interval: [9.74 9.82].
```

The two strategies can be combined to obtain the Augmented IPW (AIPW) estimator.

$$\widehat{ATE}_{AIPW} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_1(\mathbf{X}_i) - \hat{\mu}_0(\mathbf{X}_i) + T_i\frac{Y_i - \hat{\mu}_1(\mathbf{X}_i)}{\hat{e}(\mathbf{X}_i)} - (1-T_i)\frac{Y_i - \hat{\mu}_0(\mathbf{X}_i)}{1-\hat{e}(\mathbf{X}_i)}\right)$$

The AIPW estimator is double robust, *i.e.* it is consistent if either the propensity score or the outcome models are consistent. More recent approaches, such as targeted learning (TMLE) [36] and debiased ML [37] also allow the use of more expressive ML models in causal effect estimation, and are available in the `DoWhy` package as off-the-shelf implementations.

14.5 CONCLUSIONS

This chapter introduces the fundamentals of causal reasoning, as well as several existing methods to perform causal data analysis and answer two key questions, of paramount importance, particularly in healthcare. The *why* questions, in search of the causal mechanisms governing the underlying process of data generation (causal discovery), and the *what if* questions, in search of a quantification of the effect of an intervention on the system (causal inference). The proposed methods have been illustrated through simple examples, with a limited number of variables involved. Current research trends focus on how to transpose the causal machinery to the unstructured, high-dimensional and multi-channels data scenario, that is now becoming typical in healthcare. This *scaling-up* challenge requires a specific framework to combine ML, a powerful tool for feature extraction and dimensionality reduction, with causal reasoning to generate actionable insights, and will be discussed in Chapter 17.

Acknowledgments. The authors would like to thank *Julie Josse* for her useful comments and suggestions on this chapter.

BIBLIOGRAPHY

[1] A. Barragán-Montero, U. Javaid, G. Valdés, D. Nguyen, P. Desbordes, B. Macq, S. Willems, L. Vandewinckele, M. Holmström, F. Löfman, et al., Artificial intelligence and machine learning for medical imaging: A technology review, Physica Medica 83 (2021) 242–256.

[2] D. C. Castro, I. Walker, B. Glocker, Causality matters in medical imaging, Nature Communications 11 (1) (2020) 3673.

[3] B. Sahiner, W. Chen, R. K. Samala, N. Petrick, Data drift in medical machine learning: implications and potential remedies, The British Journal of Radiology (2023) 20220878.
[4] J. Pearl, Causality, Cambridge university press, 2009.
[5] J. Pearl, D. Mackenzie, The book of why: the new science of cause and effect, Basic books, 2018.
[6] D. B. Rubin, Estimating causal effects of treatments in randomized and nonrandomized studies., Journal of educational Psychology 66 (5) (1974) 688.
[7] G. W. Imbens, D. B. Rubin, Causal inference in statistics, social, and biomedical sciences, Cambridge University Press, 2015.
[8] T. S. Richardson, J. M. Robins, Single world intervention graphs (swigs): A unification of the counterfactual and graphical approaches to causality, Center for the Statistics and the Social Sciences, University of Washington Series. Working Paper 128 (30) (2013) 2013.
[9] J. Pearl, Trygve haavelmo and the emergence of causal calculus, Econometric Theory 31 (1) (2015) 152–179.
[10] C. Glymour, K. Zhang, P. Spirtes, Review of causal discovery methods based on graphical models, Frontiers in genetics 10 (2019) 524.
[11] A. R. Nogueira, A. Pugnana, S. Ruggieri, D. Pedreschi, J. Gama, Methods and tools for causal discovery and causal inference, Wiley interdisciplinary reviews: data mining and knowledge discovery 12 (2) (2022) e1449.
[12] G. P. Saptawati, B. Sitohang, Hybrid algorithm for learning structure of bayesian network from incomplete databases, in: IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005., Vol. 1, IEEE, 2005, pp. 741–744.
[13] Y. Zheng, B. Huang, W. Chen, J. Ramsey, M. Gong, R. Cai, S. Shimizu, P. Spirtes, K. Zhang, Causal-learn: Causal discovery in python, arXiv preprint arXiv:2307.16405 (2023).
[14] P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, Causation, prediction, and search, MIT press, 2000.
[15] P. L. Spirtes, C. Meek, T. S. Richardson, Causal inference in the presence of latent variables and selection bias, In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (1995) 499–506.
[16] D. Colombo, M. H. Maathuis, et al., Order-independent constraint-based causal structure learning., J. Mach. Learn. Res. 15 (1) (2014) 3741–3782.
[17] J. RAMSEY, Adjacency-faithfulness and conservative causal inference, Proc. Conf. on Uncertainty in Artificial Intelligence (UAI-06) (2006) 401–408.
URL https://cir.nii.ac.jp/crid/1572543026112772736
[18] P. Spirtes, An anytime algorithm for causal inference, in: International Workshop on Artificial Intelligence and Statistics, PMLR, 2001, pp. 278–285.
[19] D. Colombo, M. H. Maathuis, M. Kalisch, T. S. Richardson, Learning high-dimensional directed acyclic graphs with latent and selection variables, The Annals of Statistics (2012) 294–321.
[20] D. M. Chickering, Learning equivalence classes of bayesian-network structures, The Journal of Machine Learning Research 2 (2002) 445–498.
[21] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, M. Jordan, A linear non-gaussian acyclic model for causal discovery., Journal of Machine Learning Research 7 (10) (2006).
[22] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvarinen, Y. Kawahara, T. Washio, P. O. Hoyer, K. Bollen, P. Hoyer, Directlingam: A direct method for learning a linear non-gaussian structural equation model, Journal of Machine Learning Research-JMLR 12 (Apr) (2011) 1225–1248.
[23] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, D. Sejdinovic, Detecting and quantifying causal associations in large nonlinear time series datasets, Science advances 5 (11) (2019)

eaau4996.

[24] D. Entner, P. O. Hoyer, On causal discovery from time series data using fci, Probabilistic graphical models (2010) 121–128.

[25] A. Hyvärinen, K. Zhang, S. Shimizu, P. O. Hoyer, Estimation of a structural vector autoregression model using non-gaussianity., Journal of Machine Learning Research 11 (5) (2010).

[26] C. W. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica: journal of the Econometric Society (1969) 424–438.

[27] R. Marcinkevičs, J. E. Vogt, Interpretable models for granger causality using self-explaining neural networks, arXiv preprint arXiv:2101.07600 (2021).

[28] S. Löwe, D. Madras, R. Zemel, M. Welling, Amortized causal discovery: Learning to infer causal graphs from time-series data, in: Conference on Causal Learning and Reasoning, PMLR, 2022, pp. 509–525.

[29] U. Hasan, E. Hossain, M. O. Gani, A survey on causal discovery methods for iid and time series data, Transactions on Machine Learning Research (2023).

[30] P. W. Holland, Statistics and causal inference, Journal of the American statistical Association 81 (396) (1986) 945–960.

[31] A. Sharma, E. Kiciman, Dowhy: An end-to-end library for causal inference, arXiv preprint arXiv:2011.04216 (2020).

[32] P. Blöbaum, P. Götz, K. Budhathoki, A. A. Mastakouri, D. Janzing, Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models, arXiv preprint arXiv:2206.06821 (2022).

[33] I. Mayer, P. Zhao, N. Greifer, N. Huntington-Klein, J. Josse, Cran task view: Causal inference (2022).

[34] M. A. Hernán, J. M. Robins, Causal Inference: What If, Boca Raton: Chapman & Hall/CRC, 2020.

[35] P. C. Austin, E. A. Stuart, Moving towards best practice when using inverse probability of treatment weighting (iptw) using the propensity score to estimate causal treatment effects in observational studies, Statistics in medicine 34 (28) (2015) 3661–3679.

[36] M. J. Van Der Laan, D. Rubin, Targeted maximum likelihood learning, The international journal of biostatistics 2 (1) (2006).

[37] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, J. Robins, Double/debiased machine learning for treatment and structural parameters (2018).