Master's Thesis
Master's Programme in Data Science

Interactive Causal Structure Discovery

Laila Melkas

June 10, 2021

Supervisor(s): Associate Professor Kai Puolamäki

Examiner(s): Associate Professor Kai Puolamäki
Doctor Suyog Chandramouli

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Laila Melkas

Työn nimi — Arbetets titel — Title

Interactive Causal Structure Discovery

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's Thesis	June 10, 2021	79

Tiivistelmä — Referat — Abstract

Multiple algorithms exist for the detection of causal relations from observational data but they are limited by their required assumptions regarding the data or by available computational resources. Only limited amount of information can be extracted from finite data but domain experts often have some knowledge of the underlying processes. We propose combining an expert's prior knowledge with data likelihood to find models with high posterior probability. Our high-level procedure for interactive causal structure discovery contains three modules: discovery of initial models, navigation in the space of causal structures, and validation for model selection and evaluation. We present one manner of formulating the problem and implementing the approach assuming a rational, Bayesian expert which assumption we use to model the user in simulated experiments. The expert navigates greedily in the structure space using their prior information and the structures' fit to data to find a local maximum a posteriori structure. Existing algorithms provide initial models for the navigation. Through simulated user experiments with synthetic data and use cases with real-world data, we find that the results of causal analysis can be improved by adding prior knowledge. Additionally, different initial models can lead to the expert finding different causal models and model validation helps detect overfitting and concept drift.

ACM Computing Classification System (CCS):
Mathematics of computing → Probability and statistics → Probabilistic representations → Causal networks
Computing methodologies → Machine learning → Machine learning approaches → Learning in probabilistic graphical models → Maximum a posteriori modelling
Theory of computation → Models of computation → Interactive computation

Avainsanat — Nyckelord — Keywords

probabilistic causal models, causal structure discovery, interactive modelling

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Acknowledgements

I am grateful to my supervisor, Kai Puolamäki, and my other colleagues in the Exploratory Data Science research group at the University of Helsinki for all the help and support during this project. Big thanks especially to Suyog Chandramouli, Rafael Savvides, and Jarmo Mäkelä who took part in the research and helped me come up with ideas for the thesis and formulate them. I would like to thank Tuomo Nieminen and Ivan Mammarella from the Institute of Atmospheric and Earth System Research (INAR) for helping me understand the data and application area as well as for their feedback and advice during the project. This work has been funded by the University of Helsinki and by the Future Makers Funding Program 2018 of the Technology Industries of Finland Centennial Foundation together with the Jane and Aatos Erkko Foundation (Artificial Intelligence for Driving R&D project).

Author's contribution

The presented work was done in collaboration with Kai Puolamäki, Suyog Chandramouli, Rafael Savvides, Jarmo Mäkelä, Tuomo Nieminen, and Ivan Mammarella, resulting in a separate publication [61]. I have contributed to all parts of the research, including formulating the research question, literature review, formulating the theory, designing the experiments, and interpreting the results. I conducted the experiments fully myself.

Contents

1. Introduction

In natural sciences, a common aim is to construct the causal model underlying the process that generated a given set of data. This modelling becomes difficult when the data is observational rather than obtained from randomised controlled trials (RCTs) [94]. Executing RCTs is often either impractical, unethical, or both. For example, studying the effect of a harmful substance on the human body would require the exposure of a group of test subjects to the studied substance which, for obvious reasons, is not allowed by (most) scientific ethics boards. Therefore, causal modelling is often based on observational data alone.

Difficulties in causal modelling stem from the number of possible causal structures increasing superexponentially as a function of the number of variables in the model [77] as well as absence of a direct link between correlation and causation [94]. A number of algorithms have been developed to find causal structures without relying on external interventions on the process [see e.g., 79]. Causal models refer, in this context, to probabilistic graphical models that can be represented by directed acyclic graphs and that are given a causal interpretation [68]. The focus of this thesis is to specifically find only structures of the models whereas the evaluation of causal effect sizes is left outside the scope.

For a given process, the problem lies in multiple causal models being almost equally good in terms of fitting the observed data. This issue is due to the large number of possible models and to the inability to produce additional data through interventions on the system. Depending on the assumptions made regarding the process, such as the functional family of the causal relations and distributions of noise in each variable, different causal structures can be found algorithmically. The existing algorithms all make some assumptions that may not fully hold for real-world data. We aim to show how adding interactivity to causal structure discovery can help understand, refine, and potentially improve the outputs of the causal discovery algorithms.

We consider settings where a domain expert's prior knowledge, combined with automated methods, could help identify a better causal model or models than by applying the algorithms alone. Taking a Bayesian approach to formulating the problem, the domain expert has some unknown and vaguely defined prior distribution for the

causal structures that generated the observed data. At the same time, the likelihood of the data under some specific causal model can be estimated with a computer. Although the expert's prior distribution for the models cannot be accessed directly, by means of interactions with the model, a maximum a posteriori (MAP) estimation of the possible causal models can be found. Interactions comprise local edits to the causal structure by adding, deleting, or reversing a single edge at a time. To enable the expert to make such decisions, they are shown how all valid interactions on a given model would affect the likelihood of the data. Likelihood and other metrics for model selection are further discussed in Chapter 4. Assuming the expert is a rational Bayesian agent, the interactions that they choose to perform lead to a local optimum of the posterior distribution over the possible causal models.

Based on the ideas above, we propose the following workflow for interactive causal structure discovery. The expert first runs a selection of causal structure algorithms on their data. The particular set of algorithms used does not matter for our approach, as it can be extended to include any algorithm that outputs a probabilistic graphical model for some data input, although the choice of inital point for the navigation can affect the final results. In fact, the expert may introduce their prior knowledge into the analysis already in selecting which algorithms to run if they have information on which assumptions that are built in the algorithms are valid for the data. After viewing the results from the algorithms, the expert selects the model that best fits their prior knowledge and the current context based on the likelihood estimates displayed for each model. Then, the model can be edited by local edits mentioned above and the expert terminates the navigation after finding a model that cannot be improved by local edits. At each step of the editing process, the changes caused by all possible edits to the current model are shown to the expert, to facilitate navigational decisions. To evaluate how well models fit the data, we apply cross-validation which, in addition to allowing efficient use of the data, helps the expert detect problems such as overfitting and concept drift.

We test our approach by simulating a user on sets of random graphs with the level of background knowledge as a parameter. Level of knowledge here simply means the probability the expert assigns to the true edges both present in and absent from the underlying model that generated the data. The user model is simple and based on an assumption of a fully rational Bayesian agent. By these simulations, we attempt to answer the following questions. (1) Does incorporating expert knowledge into the causal structure discovery process improve the results? (2) How does the expert's level of knowledge affect the results? (3) Does the sample size have an effect on the results with and without expert knowledge? and (4) Is it useful to have multiple different initial points for the navigation?

In addition to the simulations, we present use cases executed with real-world data. We use a carbon dioxide flux data set measured in Hyytiälä [59] which is freely available online. Two domain experts use our approach to find causal structures for the data set that fit their prior knowledge and the data. We also show with a separate use case how the navigation proceeds when the user has no prior knowledge of the process that generated the data. With these use cases, we examine whether incorporating expert knowledge into the search process produces results that are different from those output by the algorithms, whether the choice of initial model for navigation affects the results, and how problems in the analysis may be detected with cross-validation.

This thesis is structured as follows. We introduce causal Bayesian networks in Section 2 including a brief discussion on the definition of causality. In Section 3, we present methods for automatic causal structure discovery. Model selection including metrics for comparing a number of models for goodness-of-fit and model validation are discussed in Section 4. We provide an overview of interactive causal structure discovery methods and causal inference in Earth system sciences in Section 5. With the exception of deriving our generalised metric for model evaluation in Subsection 4.2, our contributions in Sections 2 through 5 consist of reviewing related literature. In Section 6, we present our approach and an example of implementing it. We lay out the experimental setup and results in Section 7. In Section 8, we discuss implications and open research questions. Finally, we conclude in Section 9.

2. Causal Bayesian Networks

Stochastic systems can be described by the joint distribution of the associated variables. As the dimensionality of the model grows, the joint distribution becomes harder to specify. However, statistical independence allows us to write the joint probability of two random variables as the product of their marginal probabilities. Through dependence relations among model variables their joint probability can thus be factorised into a product of the conditional probabilities of each variable.

Depending on the context, p refers to either the probability density function of a continuous random variable or the probability mass function of a discrete random variable. Variables and nodes are denoted by capital letters $A, B, X, Y, \ldots$ and sets of variables, as well as data matrices, by bold capital letters, such as $\mathbf{Z}$. Conditional independence of two variables given a set of conditioning variables is denoted by $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ and conditioning on an empty set can be written either as $X \perp\!\!\!\perp Y \mid \emptyset$ or $X \perp\!\!\!\perp Y$.

Probabilistic graphical models offer a compact means to encode independence relations between model variables along with the conditional distributions needed to compute the factorised joint probability [e.g., 3, 50]. To describe their use in the causal context, the terminology and definitions for graphs are introduced in Subsection 2.1 and Bayesian networks are presented in Subsection 2.2. Finally, the concept of causality especially in the context of Bayesian networks is discussed in Subsection 2.3.

2.1 Graph Terminology

Graphs provide an intuitive means of representing relationships between a model's variables by visualising statistical independence and dependence relations explicitly which, in turn, provides a factorisation for the joint distribution of the variables [50]. A graph G is defined by pair (V, E) where V denotes the set of vertices or nodes and E the set of arcs or edges between them [e.g., 3, 50, 68]. An edge between two variables $X, Y \in V$ can be *directed*, $X \to Y$ or $X \leftarrow Y$, *undirected*, $X - Y$, or *bidirected*, $X \leftrightarrow Y$. We focus only on graphs with directed and undirected edges. *Non-adjacency* of two nodes, nodes without any type of edge between them, is denoted by $X \not- Y$.

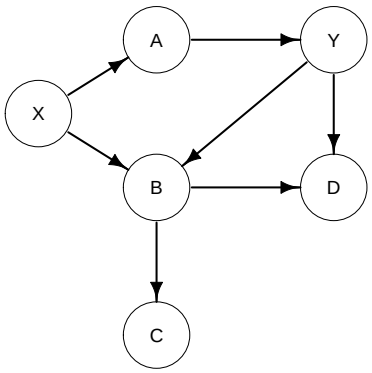


Figure 2.1: An example of a directed acyclic graph, DAG.

Directed edges are oriented from a *parent* to a *child* and two nodes with an undirected edge between them are called *neighbours*. If the graph is given a causal interpretation, parent nodes equate to causes and children to effects. The notation $\mathrm{Pa}(X)$ for *parent set* refers to the set of all of the parent nodes of X and $\mathrm{Pa}_G(X)$ is used to highlight the parent set of X in graph G. For example, the parent set of node B in the graph in Figure 2.1 is the set $\mathrm{Pa}(B) = \{X, Y\}$. A directed edge is *incoming* with respect to node X if it is oriented towards X, otherwise it is called *outgoing*. *Adjacent* nodes have either an directed or undirected edge between them. For a node X, other parents of its children that are not adjacent to X are its *spouses*. In the example graph in Figure 2.1, X and Y are spouses as they have a common child in B but are not adjacent. In *directed graphs*, all of the edges are directed and, conversely, all of the edges are undirected in *undirected graphs*. *Partially directed graphs* may contain both directed and undirected edges. Removing all of the edge orientations of a directed graph but none of the edges or nodes results in the *skeleton* of the original directed graph.

A *path* U between nodes X and Y consists of a sequence of nodes with edges between them, either undirected or directed towards Y [50]. If all of the edges on a path are directed from the first node towards the end node, the path is called a *directed path*. *Trails* may additionally contain edges that are oriented towards the first node of the sequence. Thus, all paths are trails but the same does not apply in reverse. The sequence $X \rightarrow A \rightarrow Y$ is a path from X to Y and the sequence $X \rightarrow B \leftarrow Y$ is a trail from X to Y in the example graph in Figure 2.1. *Connected* nodes have at least one trail between them and *disconnected* nodes have none. In *connected graphs* such as the one used as an example, every pair of nodes X and Y is connected by a trail. *Complete graphs* contain edges of any type between all of the nodes: all pairs of nodes X and Y are adjacent. Conversely, there are no edges in an *empty graph*. A *cycle* in a directed graph is defined as a directed path from any node to itself. *Directed acyclic*

graph (DAG) is a directed graph without any cycles. The nodes of any DAG can be sorted in a *topological ordering* which is an ordering of $X_1, \ldots, X_n \in V$ such that if $X_i \to X_j$ then X_i precedes X_j. In other words, no edges exist to a node X from any node Y which succeeds X in the topological ordering. The order is not necessarily unique: both X, A, Y, B, C, D and X, A, Y, B, D, C are valid topological orders for the example graph. Because no directed paths exist from C to D or from D to C, their mutual ordering does not matter.

A trail U with directed edges may contain three types of three-node structures in terms of the orientation of the incoming and outgoing edges of the middle node of the structure [68]. First, in a *chain* on trail U, the middle node, referred to as a *mediator*, has one incoming and one outgoing edge on U, such as $X \to A \to Y$ or $Y \leftarrow B \leftarrow C$ in Figure 2.1. Second, structures where both of the edges are directed away from the middle node, for example, $C \leftarrow B \to D$, are called *forks*. The middle node of a fork is referred to as a *confounder* or a *common cause* of the other two nodes in a causal context. Three nodes with exactly two edges between them are referred to as a *v-structure* if the edges are oriented towards the middle node, such as $B \to D \leftarrow Y$. If B and Y were linked by an edge, the three nodes would not form a v-structure but a structure with a *collider*. The term collider or, sometimes, *unshielded collider* is used for nodes that have at least two incoming edges from two other, non-adjacent nodes. If the two edges are not directed, the structure is called an *unshielded triple*. B is a collider in the graph in Figure 2.1 as the graph contains two nodes X and Y such that $X \to B \leftarrow Y$ and $X \neq Y$. D is not a collider in the graph, as its two parents B and Y are adjacent, althoug+h it is a collider on the path $X \to B \to D \leftarrow Y$. Note that for any of the structures, the middle node may have other edges to or from nodes that are not on U such as the node B in the example of a v-structure $X \to B \leftarrow Y$ with two outgoing edges to C and D.

Node X is an *ancestor* of node Y if a directed path exists from X to Y but not from Y to X as is true in the running example. In this case, Y is a *descendant* of X and the *non-descendants* of node X comprise all of those variables that are not descendants of X, thus ancestors of X and nodes unconnected to X. In the example graph, the nodes X, A, and Y are both the ancestors and the non-descendants of B and its descendants are C and D. The only two nodes of which neither is an ancestor of the other are C and D. Nodes always belong in the set of descendants of their parents and in the set of ancestors of their children. *Root nodes* have no parents and *sink* refers to a node without any children. In some sources, sink refers to the end node of a directed path [94]. X is the only root node in the example graph and both C and D are sinks.

2.2 Bayesian Networks

Bayesian networks (BN) are probabilistic graphical models represented by a DAG G which is combined with conditional probability distributions for each of the model variables given their parents in G [66]. A BN is often defined for a set of discrete variables by the factorisation of the joint distribution of the variables [3, 15, 68], although the same concepts can be extended to cover continuous variables [94]. As mentioned above, a graph structure can be used to encode information about conditional independence relations between the model variables. The amount of information regarding the model's joint distribution P directly available from DAG G depends on the conditions satisfied by P with regard to G. If a probability distribution P factorises according to a DAG G, G *represents* or is *compatible* with P [68].

Distribution P over variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ satisfies the *local Markov condition* with regard to G if and only if every node X_i is conditionally independent of its non-descendants given the set of its parents $\mathrm{Pa}(X_i)$ [68, 94]. In causal terms, the condition is referred to as the *causal Markov condition* and is defined as any variable being independent of all other variables except its effects, direct and indirect, and direct causes conditional on its direct causes. This condition implies that P can be factorised as

$$p(\mathcal{X}) = \prod_{i=1}^{n} p(X_i \mid \mathrm{Pa}(X_i)). \tag{2.1}$$

If P can be factorised as above, the local Markov condition is implied with regard to G. According to the local Markov condition, the example graph in Figure 2.1 implies the factorisation $p(X)p(A \mid X)p(Y \mid A)p(B \mid X, Y)p(C \mid B)p(D \mid B, Y)$. The condition does not, however, imply all of the independence relations in a graph. For example, consider a graph with three variables X, Y, Z, and only one edge $X \to Y$. According to the local Markov condition, Y is independent of Z given X although Y and Z are independent conditional on an empty set as seen from the factorisation implied by the graph: $p(X, Y, Z) = p(X)p(Y \mid X)p(Z)$.

All of the independence relations represented by a DAG G can be found by the concept of *d-separation* where the letter d comes from the word directional [106]. A trail between variables X and Y is *active* given a set of variables $\mathbf{Z}$ if none of the non-colliders on the trail belong to $\mathbf{Z}$ and, for each collider on the trail, either the collider or one of its descendants belongs to $\mathbf{Z}$. *Blocking* a trail refers to modifying the conditioning set $\mathbf{Z}$ by adding or removing variables from it so that the trail is no longer active. Once a trail is inactive given a set Z, Z is said to block the trail. Two sets of variables $\mathbf{X}$ and $\mathbf{Y}$ are defined as d-separated by a separate set of variables $\mathbf{Z}$ if there are no active trails between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ [e.g., 3, 50, 68, 94]. In other

words, the set $\mathbf{Z}$ blocks every trail from $\mathbf{X}$ to $\mathbf{Y}$. The conditioning set $\mathbf{Z}$ can be referred to as a *separating set* for a pair of variables when conditioning on $\mathbf{Z}$ d-separates the pair. If two sets of variables are not d-separated, they are *d-connected*. D-separation links to conditional independence: if two variables X and Y are d-separated by $\mathbf{Z}$ in graph G, then they are conditionally independent in all of the distributions P that are compatible with G [68]. Conversely, if the variables are d-connected by $\mathbf{Z}$, then they are conditionally dependent given $\mathbf{Z}$ in at least one distribution P compatible with G. The implication only applies to one direction and conditional independence of two variables in a distribution P does not imply their d-separation in a graph with which P is compatible, nor does their dependence imply d-connection.

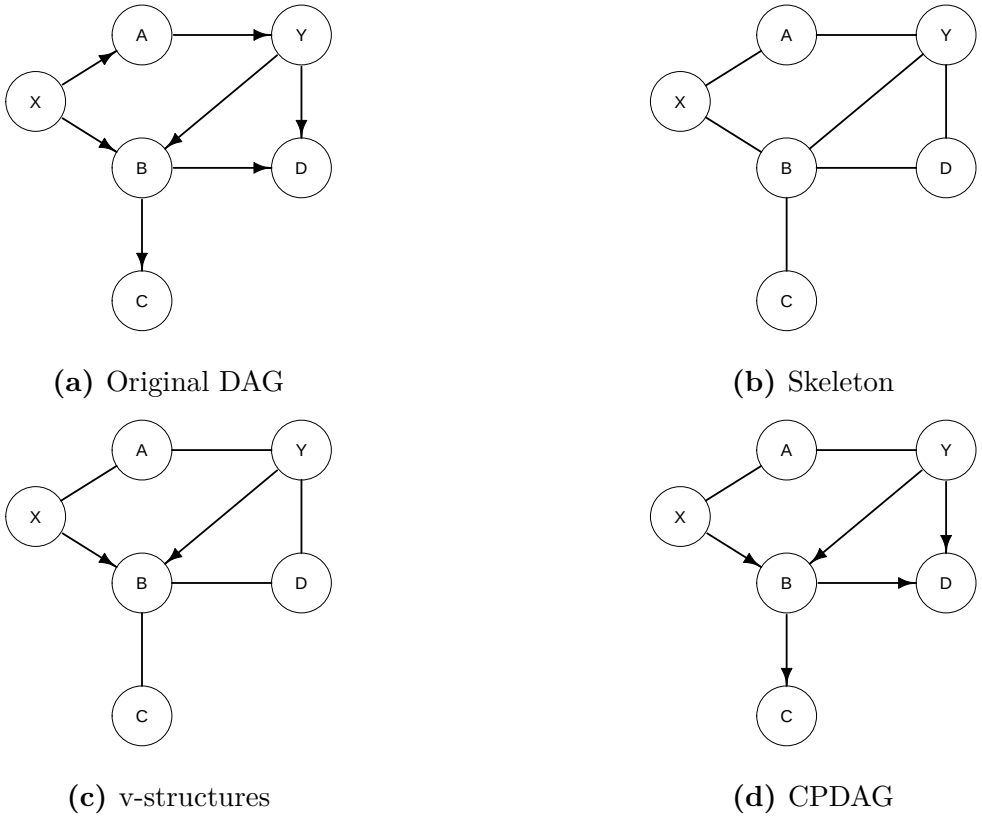
In the running example, trail $X \rightarrow A \rightarrow Y$ is active when no variables are conditioned on, rendering the two variables d-connected given an empty set. Adding A to the conditioning set blocks the trail and, as the other trails between X and Y are blocked by the collider B which is not conditioned on, introduces d-separation of X and Y. With A in the conditioning set, adding any other variable would unblock one of the trails, breaking the d-separation. As B is not conditioned on, the trail between X and D is active. Being a collider, D blocks trail $X \rightarrow B \rightarrow D \leftarrow Y$ but conditioning on it would unblock the trail. Y would thus be reachable from X if D was added to the conditioning set. Conditioning on B would have an equivalent effect by unblocking trail $X \rightarrow B \leftarrow Y$. Because C is a descendant of B, adding it to the conditioning set unblocks the same trail as conditioning on B directly.

If we know the set of conditional independence relations present in distribution P, a DAG G that is compatible with P can be built with that information. The graph representation of a distribution need not be unique, however. Consider a distribution with three variables X, Y, and Z. If the only conditional independence relation in the distribution is given as $X \perp\!\!\!\perp Y \mid Z$, the skeleton of the graph is $X - Z - Y$, as the pairs X, Z and Z, Y are not conditionally independent given any subset of the other variables. Altogether four possibilities exist to direct the two edges: $X \rightarrow Z \rightarrow Y$, $X \leftarrow Z \leftarrow Y$, $X \leftarrow Z \rightarrow Y$, and $X \rightarrow Z \leftarrow Y$. All but the last one, $X \rightarrow Z \leftarrow Y$, imply the same conditional independence relation, $X \perp\!\!\!\perp Y \mid Z$. A set of DAGs that represent the same conditional independence relations is called a *Markov equivalence class* or an *equivalence class* and two DAGs belonging to the same class are *Markov equivalent* [3, 94]. All DAGs within an equivalence class contain the same d-separation relations [94]. Two DAGs belong to the same equivalence class if they have the same skeleton and the same set of v-structures [107]. DAGs in the same class are *likelihood equivalent* if the data set meets the assumptions of linear relations and normally distributed noise [36, 94]. In such a case, the DAGs in the equivalence class cannot be separated from each other based on data alone as the model likelihood is

equal for all of the class members. The graph $X \rightarrow Z \leftarrow Y$ is the only member of its equivalence class whereas the other three graphs with the same skeleton form one equivalence class together as they have the same empty set of v-structures.

A Markov equivalence class can be uniquely represented by a *completed partially directed acyclic graph* (CPDAG) which can contain both directed and undirected edges [10]. The process of forming a CPDAG from a DAG G is displayed in Figure 2.2. The first step is to find the skeleton of the original graph, which for the running example is shown in Figure 2.2b. In the second stage shown in Figure 2.2c, all of the edges that belong to a v-structure in G are oriented with the same orientations as they have in graph G. The example graph contains only one v-structure, $X \rightarrow B \leftarrow Y$. Even though $B \rightarrow D \leftarrow Y$ is a structure with a collider, it is not unshielded because B and Y are adjacent and, therefore, is not a v-structure. Finally, as many of the remaining edges as possible are oriented by enforcing two rules: no new v-structures must be created and no cycles introduced. The arrow $B - D$ must be oriented towards D in order to avoid introducing a new v-structure $X \rightarrow B \leftarrow D$ and for the same reason the arrow $B - C$ is oriented towards C. After the two edges have been oriented, it can be seen that the arrow $D - Y$ must be directed towards D. Otherwise, the graph would contain a cycle $B \rightarrow D \rightarrow Y \rightarrow B$. Regardless of the orientation of the two edges that remain undirected, no new v-structures or cycles are created, unless both edges were oriented towards A. The resulting CPDAG in Figure 2.2d is a unique representation of the equivalence class of the original DAG.

A common assumption in Bayesian networks is that of *faithfulness* for a given distribution P and graph G [50]. The faithfulness assumption states that the conditional independence of variables X and Y given a separate set of variables $\mathbf{Z}$ implies the two are d-separated in G by $\mathbf{Z}$. Faithfulness often holds, as introducing *unfaithful* independence in a distribution requires precise fine-tuning of the model parameters. In some practical applications, such fine-tuning is performed purposefully to achieve independence of two variables. For example, the air temperature inside a fridge depends on the outside temperature but as the impact is actively counteracted, the two quantities would statistically be independent. In this case, the distribution would include a statistical independence between two variables that would not be d-separated in the graph representing the process that generated the data. Unfaithful relations thus can be created by designing a system to artificially introduce independence between two or more quantities. Although most systems adhere to the faithfulness assumption, problems can arise in detecting weak dependence between variables. Weak dependence can be misclassified as independence leading to errors in the graph structure. With the d-separation criterion and the assumption of faithfulness, distribution P and compatible graph G contain the same set of conditional independence relations.



(a) Original DAG **(b)** Skeleton

(c) v-structures **(d)** CPDAG

Figure 2.2: Finding the completed partially directed acyclic graph for the Markov equivalence class of a given DAG.

2.3 Causality

As the famous phrase states, correlation does not imply causation. Statistical relationship between a pair of variables is insufficient to determine a causal connection between the two [e.g., 68]. A classic example that highlights this problem concerns the correlation between the amount of ice cream consumed and the number of drownings. Clearly, eating ice cream does not cause one to drown nor does the causal relation hold in the opposite direction. In this example, the problem arises from missing variables, such as air temperature, that affect the probability of both events: both amount of ice cream eaten and number of drownings tend to increase in warm summer months. Correlation between two variables without a direct causal link between them is referred to as *spurious correlation* [89].

Simpson's paradox describes a phenomenon where the sign of the correlation coefficient is reversed for the full data set as opposed to separate subsets of the data [7, 90]. As a fictional example, consider a data set collected by asking a number of people their age, how much they exercise on average within a given time frame, and their cholesterol level. The example is visualised in Figure 2.3. Inspecting the data set as a

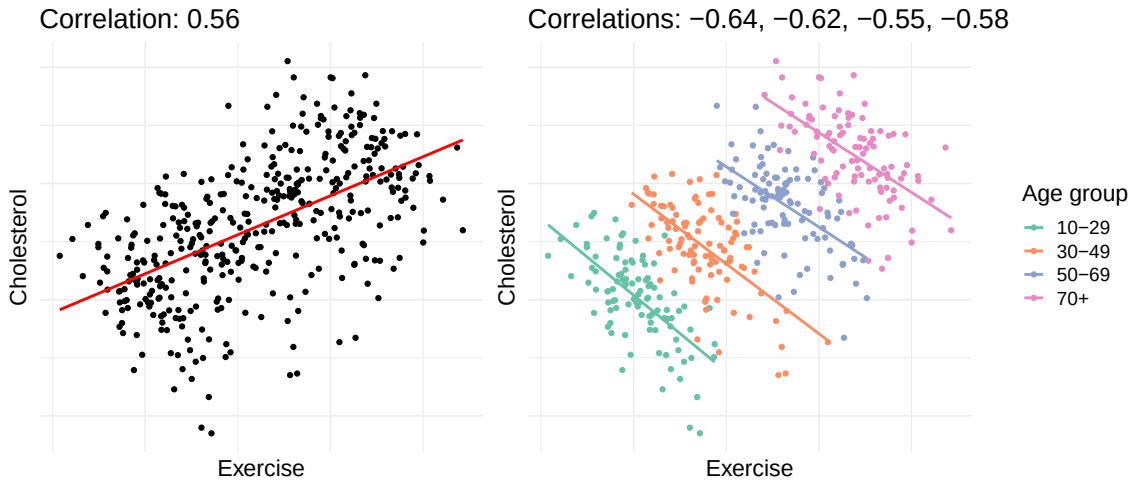


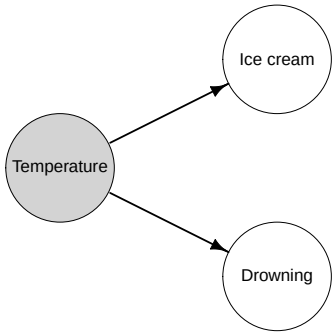
Figure 2.3: An example of Simpson's paradox.

whole, the conclusion would be that higher amounts of exercise correlate with higher cholesterol values which seems counter-intuitive. Once the different age groups are analysed separately, on the other hand, the results support the prior belief of beneficial effects of exercise on cholesterol levels. The reason for this "paradox", in this case, is that age correlates with higher amounts of both exercise and cholesterol or, in other words, age acts as a confounder of the two other variables. Thus, the phenomenon serves as a example of the importance of including all relevant variables in the analysis. Once confounding variables are accounted for, true relationships between the variables can be detected.

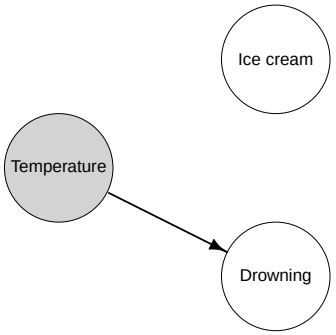
The discrepancy between causation and correlation raises the question of how causal relationships are defined. Even the whole existence of causality as a concept that is separate from correlation has been disputed with the argument that science describes past events rather than defines necessary consequences of a sequence of actions or perceptions [70, 62]. By this definition, causation is equated to perfect correlation and spurious correlations are explained away by lack of data. Other definitions that separate causality from correlation have been presented in various fields, including statistics and philosophy.

In terms of propositional logic, causality can be defined as a deterministic relation with a set of laws that determine the values of atomic propositions [88]. If a minimal set of laws L_a required to determine a proposition a contains as a proper subset a minimal set of laws L_b that determine proposition b, proposition b is said to have causal precedence over a relative to L_a. In other words, b is said to cause a. Another definition for deterministic causality states that an event a is a cause of event b if a and b both occur and b would not have occurred without a [55].

Granger causality defines a causal relation between two time series a and b as



(a) Causal structure with air temperature as a latent variable.



(b) Causal structure after intervening on eating ice cream.

Figure 2.4: An example of performing an intervention to detect causal relations.

the ability to use temporally preceding values of one time series to predict temporally later values of the other [30]. Thus, a is said to cause b if a helps predict values of b and a precedes b temporally. The definition includes a stochastic noise term, thus allowing for the found relationships to be interpreted probabilistically. However, as Granger causality only relies on statistical association, it has been said to serve to find forecasting rather than causal relations [31].

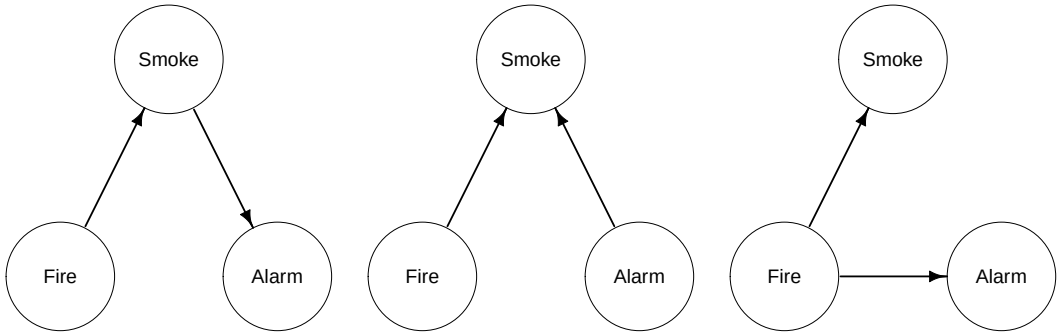
To separate statistical dependence from a causal relationship, the difference between *observations* and *interventions* has been used to help define causality [68]. Conditional probabilities of events, or variables, can be found by observing the values of the variables. How the value of one variable affects the value of another cannot be determined based on observation alone due to the possibility of confounding effects of other *latent* or unconditioned variables, as exemplified by the link between eating ice cream and number of drownings. Intervening on the suspected cause while controlling for all other variables except the suspected effect, on the other hand, allows the detection of causal relationships between the two.

Randomised controlled trials rely on essentially the same argument [95]. Using sufficiently large random samples from the population and changing the value of the variable of interest, for example medical treatment, equates to an intervention while controlling for possible confounders [68]. An interventional notation for probabilities with the *do-operator* has been introduced, $p(Y \mid \mathrm{do}(X))$, together with a set of inference rules referred to as *do-calculus* [67]. The notation differs from that of conditional independence, $p(Y \mid X)$, to highlight the interpretation of intervening on X by actively setting X to some value. Interventions do not suffer from confounders because setting X to a specific value severs the links between X and its causes as the causes no longer affect the value of X. However, the suspected effect can still be caused by other variables.

Continuing with the simple ice cream example, Figure 2.4 visualises the change in causal relations when a variable is intervened on. The left panel shows the assumed true causal structure for the three variables of which air temperature is unobserved. From the structure, it can be seen that eating ice cream and drowning are statistically dependent without interventions or otherwise controlling for the air temperature. If we, for example, know the number of drownings our probability for the amount of ice cream consumed is adjusted accordingly: a large number of drownings raises the probability we assign for the current season to be summer which, in turn, raises our probability that large quantities of ice cream is eaten. Explicitly setting the value of variable ice cream to x, on the other hand, breaks the causal effect of temperature on ice cream, shown in the right panel of the figure. As the value is manually set, the temperature no longer has an impact on it. Because a pair of variables with a common cause are unconditionally dependent, either the common cause must be conditioned on or the causal relations must be broken to detect the non-adjacency of the two.

In addition to statistical association and intervention, a third, more abstract approach to identifying causal relationships is the use of *counterfactuals* [68]. Counterfactuals are defined as statements or queries of the form "what would have happened", they concern the possible consequences of events that did *not* occur. Either deterministic or stochastic approach can be assumed when counterfactuals are used to define causality. As reasoning with counterfactuals requires the imagining of situations contrary to what is observed, at least the current technological approaches cannot apply this approach to automatically identify causal relationships. Consequently, including people in the process is necessary to enable the detection of causal structures. This notion holds especially when the available data is purely observational as the measurements do not result from one or more interventions.

In the context of causal inference, having data for the variables relevant to the process under inspection is essential for valid analysis as highlighted by a few simple examples above. Including common causes is needed to account for possible confounding effects when interventions are not possible for ethical or practical reasons. Whether all effects of the variables are included does not matter in terms of the validity of the causal structure. The assumption of *causal sufficiency* guarantees that all common causes of two or more variables in set V are included in V [94]. Furthermore, including latent common causes of two or more variables in the model does not satisfy the assumption but they additionally have to be measured or have constant value. In the ice cream example, either the air temperature would have to be measured or we could look at a data set where it was constant. For example, if only measurements for days with an average temperature of 15 degrees Celsius were analysed, the absence of a causal relation between the two measured variables could be identified as the temperature



(a) True model **(b)** SHD to true model: 1 **(c)** SHD to true model: 2

Figure 2.5: Example of performing comparisons among causal models with the Structural Hamming Distance (SHD).

would not affect their values.

Distributional comparison among models with regard to a given data set is discussed in Subsection 4.2. When a ground truth model is available, identified causal structures can be evaluated against the true model at structural level without accounting for model parameters and data distribution. For example, the *structural Hamming distance* (SHD) between two graphs represents the number of *edits* required to transform one graph into the other [16, 101]. Each edit comprises adding, deleting, or removing an edge. The distance is computed by summing together how many missing edges, extra edges, and edges with an opposite orientation there are in a graph compared with another graph. With SHD we can perform comparisons among both DAGs [16] and completed partially directed acyclic graphs (CPDAGs) [101]. Although SHD provides a useful metric for similarity between two graphical models, its application is less straightforward when the studied models have a causal interpretation.

Consider an example of the causal structure between three variables "Fire", "Smoke", and "Alarm" [69]. In the true causal structure for the process, fire causes smoke and smoke causes a smoke alarm to go off, as displayed in Figure 2.5a. Now, consider two causal models with SHD of one and two to the true model, shown in Figures 2.5b and 2.5c, respectively. The model (b) with fire and alarm as causes for smoke is more similar to the true model as measured with SHD than the model (c) with fire as a cause for both smoke and alarm. However, it could be argued that model (c) somehow represents common knowledge of the causal relationships between the three variables more truthfully than model (b), although it misses the causal link between smoke and alarm completely.

As shown in the example, SHD cannot be used blindly to determine which of a set of causal models represents the true causal relations best even when the ground truth is known. As a combination of missing, extra, and misoriented edges, the use of

SHD for comparison can result in loss of information. Nonetheless, SHD is a common metric for comparing a number of model structures to a known ground truth [e.g., 40] and found useful especially in conjunction with other model metrics [16].

3. Causal Structure Discovery

The elicitation of causal structures from domain experts is time-consuming and uncertainty in the elicited models cannot be easily measured [102]. Furthermore, all of the causal connections in a given process can be unknown even to the experts. For these reasons, automated methods for *causal structure discovery* (CSD) have been long developed and researched. Unless otherwise specified, the algorithms presented in this chapter assume faithfulness and causal sufficiency as defined above in Subsections 2.2 and 2.3, respectively. Possible other assumptions required by each algorithm are stated explicitly.

We use a synthetic data set with five variables to demonstrate how the discussed algorithms work. The data set contains a hundred data points sampled from the following linear set of structural equations:

$$A := E_A \tag{3.1}$$

$$B := E_B \tag{3.2}$$

$$C := A - B + E_C \tag{3.3}$$

$$D := B + E_D \tag{3.4}$$

$$E := 0.5 * C + E_E \tag{3.5}$$

E_i denotes the noise term of variable i. Each of the noise terms follows a zero-mean Gaussian distribution with unit variance.

Constraint-based CSD algorithms that we discuss in Subsection 3.1 use statistical tests of conditional independence to infer causal relationships between variables. Pairs of variables that are found to be dependent conditional on any subset of the other model variables are joined together with an edge. The term constraint-based refers to using conditional independence relations as constraints on the set of possible graphs for a given data set. We introduce *score-based* algorithms in Subsection 3.2. They use a scoring metric to rank a heuristically chosen subset of possible causal graphs and returning the highest-ranking model. For example, models can be scored with BIC or log-likelihood of the model. We discuss scoring metrics for models in general later in Subsection 4.2. In Subsection 3.3, we present algorithms that use *structural equation*

models to find causal structures.

3.1 Constraint-based Algorithms

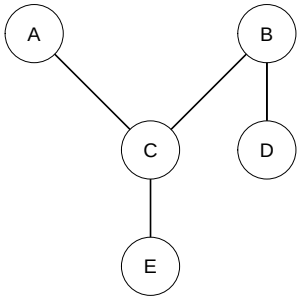
PC algorithm [93], which is named after its developers Peter Spirtes and Clark Glymour, works in two phases. The first phase consists of identifying the skeleton, the undirected graph that underlies a directed causal model. Beginning from a complete undirected graph, each edge is tested for independence given a set of conditioning variables that increases in size by one in each iteration. In the first iteration, pairs of variables are tested for independence given an empty conditioning set. For a pair of variables A and B, the conditioning set is a subset of either the neighbours of A or the neighbours of B, excluding A and B themselves. If neither A nor B has sufficient neighbours for a conditioning set of the given size that is determined by the current iteration, the two variables are interpreted to be dependent and, thus, adjacent. Whenever an independence is discovered, the set of conditioning variables is stored as the separating set for the pair of independent variables. Once none of the variables have more neighbours than the size of the conditioning set, the first phase terminates.

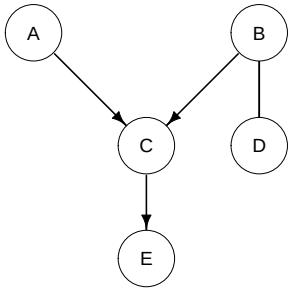
In the second phase, the separating sets are used to find v-structures. V-structures are discovered by identifying sets of three variables with exactly two edges between them and where the separation set of the two non-adjacent variables does not include the third variable. The edges in these structures are oriented towards the node with an edge to each of the other two variables. Once all of the v-structures have been discovered, as many of the remaining undirected edges are oriented as possible. An orientation can be locked if the opposite orientation would create a new v-structure or a cycle. Edges are oriented by applying these rules until no changes can be made. The algorithm outputs a completed partially directed acyclic graph (CPDAG), the unique graph representation of a Markov equivalence class, which can still contain unoriented edges.

Example outputs from the two phases of running PC are displayed in Figure 3.1. The first phase produces the skeleton of the causal model as well as the separating sets for all of the pairs of variables that are adjacent. For the final result, as many of the edges are oriented as possible adhering to the rules detailed above. The example graph contains four unshielded triples: $A - C - B$, $A - C - E$, $B - C - E$, and $C - B - D$. The only one of these where the separating set of the two non-adjacent nodes does not include the middle node is $A - C - B$. Thus, the two edges are oriented as $A \rightarrow C \leftarrow B$. Because the separating sets imply no v-structures for the other unshielded triples, the edge between C and E must be oriented towards E. Finally, the edge $B - D$ is left unoriented as either direction is compatible with the

Vars	SepSet
A, B	
A, D	
A, E	C
B, E	C
C, D	B
D, E	B

(a) Separating sets and skeleton





(b) CPDAG

Figure 3.1: Results from the two phases of the PC-algorithm.

identified conditional independence relations. The resulting equivalence class contains the underlying true model in which the final edge is oriented as $B \rightarrow D$.

Variations of PC include, for example, the conservative PC (CPC) [76] and majority rule PC (MPC) [12] both of which only direct edges in a v-structure if the unshielded triple is determined *unambiguous*. The two algorithms differ in terms of the definition of unambiguity. Detection of v-structures is performed by testing the independence of the two non-adjacent nodes in an unshielded triple given all of the subsets of their possible parents. The added conditional independence tests then result in one or more separating sets for the two nodes. In CPC, an unshielded triple $A - B - C$ is unambiguous if B is included in either all or none of the possible separating sets between A and C. On the other hand, MPC labels the unshielded triple unambiguous if the number of sets that B is an element of is not exactly half of the number of separating sets. Both algorithms categorise any remaining triples, which have not been labelled unambiguous, as ambiguous. PC-Stable has been developed to overcome the limitation of order-dependency of the original algorithm and is based on using the adjacency lists of the nodes at the beginning of each iteration for levels of adjacency in the skeleton discovery phase [12]. In addition, the orientation phase is guaranteed to be order-independent with MPC's strategy for finding v-structures.

The assumption of linearity in PC can be relaxed by selecting conditional independence tests suitable for detecting non-linear relationships between variables. One such extension is the kPC algorithm that applies *kernel methods* for independence detection [99]. However, the approach does not scale well in terms of sample size or the number of variables. Another non-parametric method for estimating non-linear dependencies, GPDC, applies *Gaussian process regression* on the data to identify independence relations [80]. Although the GPDC method facilitates discovering non-linear relationships, it suffers from a lower detection power for linear relationships with small sample sizes. Both kPC and GPDC inflict a higher computational cost than linear de-

pendency estimation such as partial correlation and Gaussian conditional independence test.

Fast Causal Inference (FCI) has been proposed as an algorithm that allows for latent variables in the model for cases where causal sufficiency cannot be guaranteed [94]. The algorithm outputs a *partial ancestral graph* (PAG) instead of a DAG or a CPDAG to enable encoding possible latent variables. In terms of edges, a PAG can contain bidirected $\leftrightarrow$ and directed $\rightarrow$ edges and edges with unknown orientation either at one $o\!\!\rightarrow$ or both $o\!\!-\!\!o$ ends. A bidirected edge implies the two adjacent variables have a latent common cause and directed edges are interpreted as in DAGs and CPDAGs, the source being a direct cause of the destination node. The symbol o at the end of an edge indicates uncertainty about whether an arrow should be drawn there. Furthermore, a variable with two uncertain edges, such as $o\!\!-\!\!oAo\!\!\rightarrow$, can be marked as non-collider $o\!\!-\!\!\underline{oAo}\!\!\rightarrow$ in which case at least one of the edges is outgoing.

Before outlining the algorithm, we provide the definition for a *possible d-separating set* of the ordered pair A, B, denoted by PD-sep(A, B). If $A \neq B$, PD-sep(A, B) contains all of those variables in $V \setminus \{A, B\}$ to which exists an undirected path U from A such that all of the variables on U except the end nodes are either colliders or possible colliders. A variable Y is a possible collider on path U if it belongs to an unshielded triple X, Y, Z on U and neither edge $X - Y$ nor $Y - Z$ is oriented away from Y. Thus, PD-sep(A, B) can differ from PD-sep(B, A).

FCI consists of four phases, the first of which equals the first phase of the PC algorithm, finding the skeleton of the causal model beginning from a complete undirected graph. The second phase comprises the identification and orientation of v-structures from unshielded triples based on the separating sets discovered in the first phase, as in PC, although no further orientations are performed. In the third phase, a PD-sep is constructed for each adjacent ordered pair of variables. If a pair of adjacent variables A and B are d-separated given any subset of either PD-sep(A, B) or PD-sep(B, A), the edge between them is removed and the subset is added to the separating set of the variables. Without an oracle available to determine true d-separation, independence tests are used to identify d-separation as in the first phase of the algorithm. Once all of the adjacent pairs are tested given their possible d-separation sets, the remaining edges are oriented in the final phase.

The orientation begins with initialising all of the edge orientations as uncertain, marked with o at each end. Then, as in the second phase, v-structures are found and oriented based on the separation sets of the variables in unshielded triples. If an unshielded triple is not a v-structure, because the middle node is included in the separating set of the two non-adjacent nodes, the middle node is marked as a non-collider on the path containing the triple. As many of the remaining edges as possible

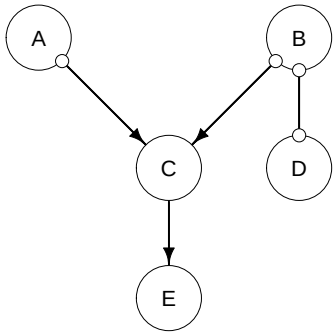


Figure 3.2: Partial ancestral graph output by FCI-algorithm.

are oriented following the rules that no new v-structures or cycles are created. Running the algorithm on the same example as the PC algorithm outputs the PAG displayed in Figure 3.2. The structure mostly resembles the one output by PC, but the algorithm is unable to determine whether there are latent common causes for the variable pairs $A - C$, $B - C$, and $B - D$.

Real-world data sets are often the result of multiple measurements at some time intervals. PCMCI is an algorithm specifically designed for detecting lagged causal relations from time series data [80]. The approach is based on combining a version of the PC algorithm with *momentary conditional independence* (MCI) to generate a method able to discover both linear and non-linear causal connections. In the first phase, PC is applied to find possible time-lagged parent sets for each of the variables. As no contemporaneous links are taken into account in this phase, all of the links can be oriented with the assumption that causes must temporally precede their effects. Let X_t reference the variable X measured at time step t. Now, any variables $X_{t-\tau}$, $\tau > 0$, with a link to variable Y_t are included in the possible parent set of Y_t.

Once the possible parent sets for all of the model variables have been found, they are used to test for causal relationships between all pairs of variables with lags between 0 and the maximum allowed time lag. The MCI test can be performed with any conditional independence test and refers to the method of testing for a causal connection between variables Y_t and $X_{t-\tau}$ given the possible parent sets of both variables retrieved in the first phase. The possible parent set of $X_{t-\tau}$ consists of the possible parent set of X_t time-shifted by τ. Given the local Markov assumption, each variable is independent of its non-descendants conditional on its parents. Because of this assumption, testing for the independence of Y_t and $X_{t-\tau}$ given the possible parent set of Y_t allows the determination of whether the two variables are causally linked. Adding the possible parent set of $X_{t-\tau}$ to the conditioning set counteracts the influence of autocorrelation in the time series. In the second phase, contemporaneous links are tested as well but they are left unoriented as the temporal ordering does not exist to help distinguish the

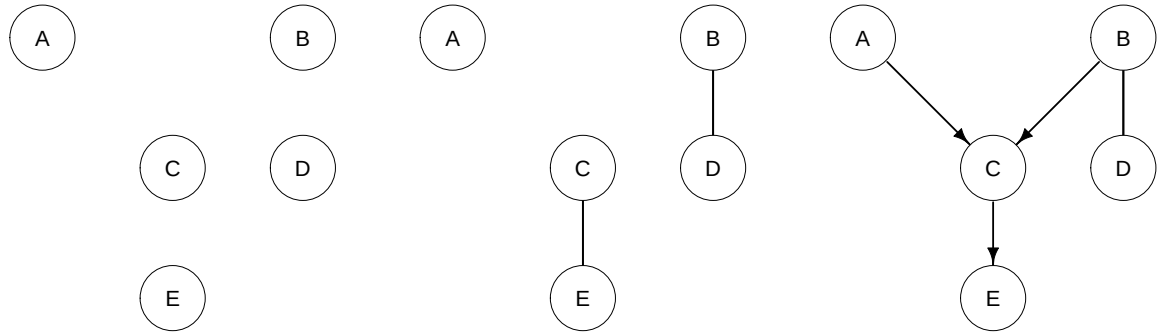
cause from the effect. An extension of the algorithm, PCMCI$^+$, has been presented to address the orientation of contemporaneous causal links [78].

The simplicity and low computational cost of the PC algorithm renders it a good choice for analysing even large sets of data if the base assumptions of linearity and Gaussianity are met. Different variants of PC provide some flexibility in terms of stability of the results and assumptions regarding the data-generating process. FCI provides a useful tool for analysing data sets when causal sufficiency is not guaranteed, although its computational complexity due to the additional independence tests in the third phase limits its application to large data sets. If the data set contains a temporal dimension, PCMCI can be applied to detect both linear and non-linear causal relationships from high dimensional time series. On the other hand, PC can be applied to large linear time series data with good results if the temporal distances between causes and effects are within the time resolution of the measurements. Using PC instead of PCMCI to analyse time series data can be justified when the data set is large because the time complexity of PCMCI is significantly higher than that of the PC algorithm.

3.2 Score-based Algorithms

Greedy equivalence search (GES) is a greedy algorithm that finds a locally optimal Markov equivalence class with respect to some *decomposable* scoring metric for which a common choice is the BIC score [11, 60]. Decomposability of the score refers to the global score of a graph being equal to the sum of the local scores of subgraphs that consist of each variable and its parents separately [3]. The log-likelihood of a graphical causal model is decomposable because the model likelihood factorises to terms of the likelihood of a node given its parents, one term for each node. Although the algorithm is based on assumptions of Gaussian noise distributions and linearity of the causal relationships, it has been found to produce reasonable results even with moderate non-linearity and non-Gaussianity [75].

Beginning from any causal graph, usually an empty graph, the algorithm works in phases. First, edges are added to the model by selecting the one that improves the graph's score most until no improvements to the score can be made by further additions. Second, edges are removed one by one with the same logic until a local optimum has been reached. Finally, in the turning phase, different orientations for the edges are tested similarly. The result from running GES on the running example beginning from an empty graph is shown in Figure 3.3. The middle graph 3.3b shows an intermediate result of the algorithm before convergence. The BIC scores displayed under each graph can only improve by decreasing in every step.



(a) BIC: 755.59 **(b)** BIC: 650.79 **(c)** BIC: 542.88

Figure 3.3: Results from running GES, beginning from (a) an empty graph, through (b) an intermediate result with final result shown in (c). Below each graph is shown its BIC score, lower is better.

All of the edits in GES are done in the space of equivalence classes and, therefore, edges are added without orientation unless the edge belongs to a v-structure or could create a cycle. The removal or turning of an edge can result in unorienting other edges that either belonged to a v-structure with the edited edge or were oriented to prevent cycles. In practice, the edits are performed on oriented edges and the resulting graph is transformed into the CPDAG representing the equivalence class that contains the partially oriented graph. The score computed for each equivalence class during the navigation is valid for every member DAG of the class due to models of an equivalence class being likelihood equivalent under the assumptions of linearity and Gaussianity of the noise distributions. Likelihood equivalence between members of an equivalence class enables the computation of the model's likelihood for the BIC score in each step by computing the estimate for any realisation of the class. Thus, when testing an edge $A - B$ for addition, it suffices to compute the score difference either for $A \rightarrow B$ or for $A \leftarrow B$.

The three phases are iterated over until the score cannot be improved by any single modification to the Markov equivalence class. The turning phase and iteration over the phases were not introduced in the original algorithm, but have been found to improve the results later [34]. GES is inefficient when node in-degree is high as it scales exponentially with regard to the size of the largest clique [11]. A *clique* in a graph is a subset of nodes all adjacent to each other, a complete subgraph. The problem in scaling can be counteracted by setting a limit to the number of parents allowed for any one node. However, the solution introduces a new problem of how to select a correct limit. A suboptimal value leads to a deterioration in the performance of the algorithm.

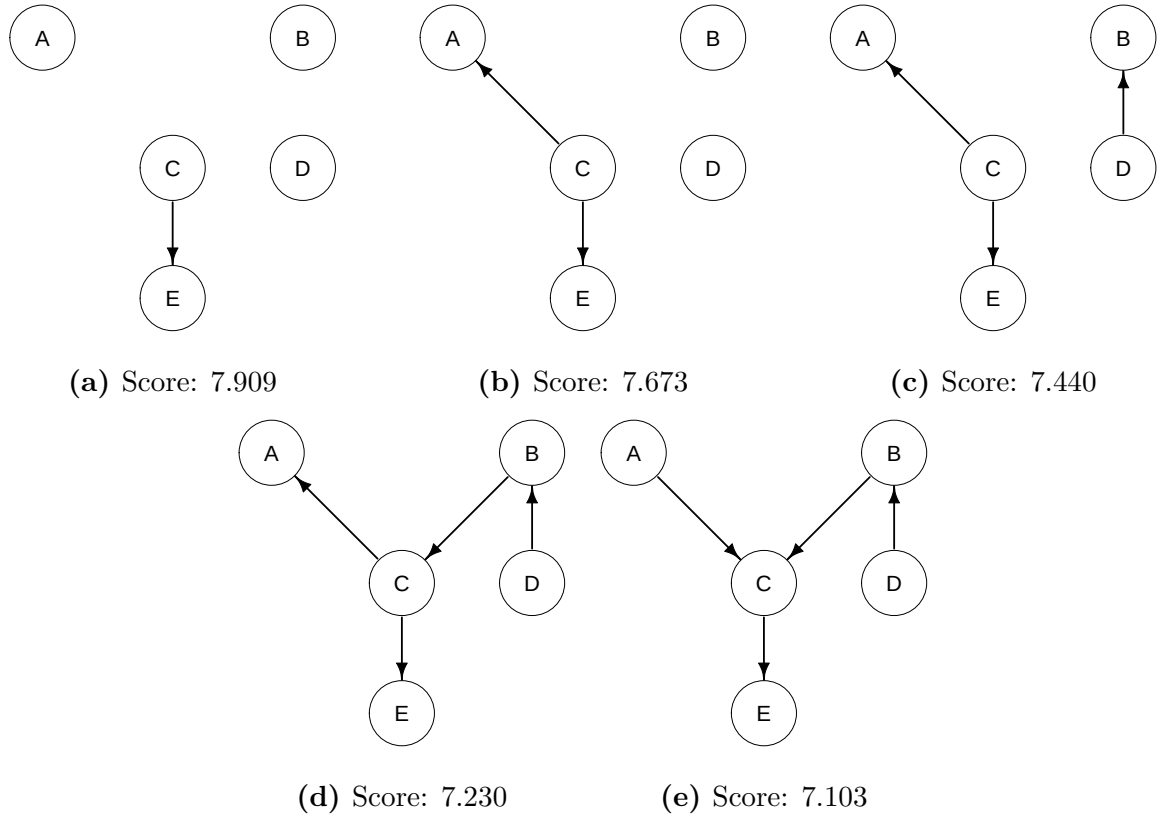
Optimisation of the GES has lead to a variant of the algorithm called the fast greedy search (FGS) [75]. Higher computational efficiency is achieved by parallelisa-

tion of the computation and by storing in a list information about all such edges whose addition in the first phase of the algorithm would improve the overall score. Keeping in memory those edges speeds up the first phase as the score differences from adding each of the absent edges need not be separately computed after every step. The decomposability of the scoring metric guarantees that the addition of an edge $A - B$ does not affect the score differences resulting from the addition of edges between nodes that are not adjacent to either A or B.

Another variant of GES, the independent multiple-sample GES (IMaGES), has been proposed to enable building aggregate causal models from multiple separate data sets [74]. Combining data gathered from multiple sources such as medical patients or measurement sites can introduce causal links between variables not present in any single data set. The IMaGES algorithm addresses this problem by estimating the log-likelihood of a graph G separately on each data set and then computing their average to estimate the log-likelihood of the aggregate model. The data sets are assumed to contain a roughly equal number of samples. Furthermore, the approach takes into account data missing from a subset of the available data sets by computing the per-data-set scores only for those edges whose end nodes are included in each data set.

A mainly score-based algorithm for CSD, fast hill-climbing (FHC) [24] resembles GES in that it applies a greedy approach to the problem by performing single edits to graphs to detect a locally optimal causal model for a given data set. Unlike GES, FHC navigates in the space of DAGs directly instead of Markov equivalence classes. The algorithm uses conditional independence tests to constrain the number of calculations for score differences, in addition to employing a scoring metric for comparing models. The search consists of one phase only during which both additions and removals of edges are considered iteratively until no edit would improve the score. Each node A is associated with a constantly updated list of nodes that are not parents of A. When computing the impact of an edit on the model's score, the two nodes at each end of the edited edge are tested for independence conditional on the parents of either node. If the nodes are found conditionally independent, neither can be a parent of the other one. Intermediate and final results from applying FHC on the example are shown in Figure 3.4. Although the final result contains an erroneous orientation in edge $D \rightarrow B$, it belongs to the same equivalence class with the true model.

For relatively low-dimensional data sets, GES has been found to perform well even with the assumptions of linearity and Gaussianity partially broken [75]. If the number of variables is in the order of hundreds or thousands, on the other hand, FGS provides a more efficient method for CSD than GES. With multiple data sources that cannot be combined without the risk of introducing new causal connections, IMaGES can be used to build an aggregate model even when all of the data sets do not contain



(a) Score: 7.909 **(b)** Score: 7.673 **(c)** Score: 7.440

(d) Score: 7.230 **(e)** Score: 7.103

Figure 3.4: Results from FHC beginning from an empty graph. Scores are proportional to BIC.

measurements for all of the model variables. Although FHC offers speed-ups for *sparse graphs*, which are graphs where nodes have few parents on average, the additional conditional independence tests reduce the achieved efficiency.

3.3 Structural Equation Models

Structural equation models (SEMs) represent causal structures by defining each variable as a function of its parents and some noise [68, 115]. Each SEM has a unique causal graph representation which contains no cycles if the model is a *recursive SEM* [e.g., 38]. In this thesis, only recursive SEMs are considered and, for simplicity, the attribute "recursive" is dropped from the term in further mentions. The noise terms are assumed to be independent of each other, resulting in a useful asymmetric relationship between a child and a parent. Because the noise terms are mutually independent and the child is defined in terms of its parents, the parent is independent of the noise term of the child, but the child is not independent of its parent's noise term. The various algorithms that use structural equation models differ in the assumptions that are made regarding the underlying causal model.

LiNGAM, an abbreviation for *linear non-Gaussian acyclic models*, applies *inde-*

pendent component analysis (ICA) to identify causal relations and causal effect sizes under assumptions of linearity of the causal relations and non-Gaussianity of the noise distributions [87]. Only linearity, non-Gaussianity, and causal sufficiency are assumed, the model is not required to satisfy faithfulness. ICA provides a method for separating the independent components of a multivariate data set given sufficient samples by taking advantage of the asymmetry between the variables stemming from the assumption of non-Gaussianity [13, 87].

Let $\mathbf{X}$ be a $n \times p$ data matrix where each of the n rows corresponds to a p-dimensional observation. Let $\mathbf{B}$ be the coefficient matrix for the model variables that is permutable to a strictly lower triangular matrix. The requirement of permutability of $\mathbf{B}$ stems from the assumption of the causal model's acyclicity. Now, a linear, non-Gaussian acyclic causal model can be defined as

$$\mathbf{X} = \mathbf{B}\mathbf{X}^T + \mathbf{E}, \tag{3.6}$$

where $\mathbf{E}$ denotes the noise matrix that consists of the independent components [87]. ICA is applied to find a decomposition of the observed data matrix $\mathbf{X} = \mathbf{A}\mathbf{E}$, where the mixing matrix is given by $\mathbf{A} = (\mathbf{I} - \mathbf{B}^T)^{-1}$. Non-Gaussianity of the noise distributions ensures that $\mathbf{A}$ is identifiable [13].

The first phase of the algorithm consists of applying ICA to estimate the mixing matrix $\mathbf{A}$ [87]. As ICA can identify the structure only up to a permutation of rows and columns [13], a few additional steps are required to find a unique causal structure. $\mathbf{A}^{-1}$ is permuted to produce a matrix $\widetilde{\mathbf{W}}$ with a diagonal of all non-zero elements and each row of $\widetilde{\mathbf{W}}$ is divided by the corresponding diagonal element. Due to estimation errors, the permutation with non-zero diagonal is not, in practice, unique for which reason $\widetilde{\mathbf{W}}$ is estimated by finding the permutation of $\mathbf{A}^{-1}$ that minimises the sum of the inverses of its diagonal elements. By subtracting the resulting $\widetilde{\mathbf{W}}$ from an identity matrix of equal dimensions, an estimate of $\mathbf{B}$ of causal effects is obtained.

To discover the causal ordering, the matrix still needs to be permuted to be as close to a lower triangular matrix as possible [87]. A number of pruning techniques can then be applied because, in practice, the result would be a full model as the estimated values are only approximately zero where the true theoretical value is zero. The DAG obtained by applying LiNGAM on the running example is displayed in Figure 3.5. As the noise distributions in the example data follow a Gaussian distribution, the algorithm performs worse than the others that rely on Gaussianity. However, even with one of the base assumptions broken, all of the true edges are found with only one edge, $E \to C$, incorrectly oriented and one extra edge.

Non-linear additive noise models (non-linear ANMs) provide a method to estimate causal models from observational data without assuming linearity or Gaussianity [40].

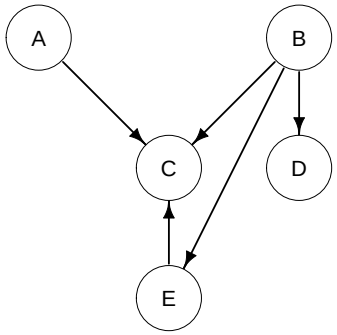


Figure 3.5: Result from LiNGAM

Causal models take the form

$$X_i = f_i(\mathrm{Pa}(X_i)) + E_i, \tag{3.7}$$

for each of the model variables X_i where E_i denotes the additive noise term, f_i is some arbitrary function for X_i, and $\mathrm{Pa}(X_i)$ is the parent set of X_i in the model. No assumptions are made regarding either the form of f_i or the distribution of E_i.

The proposed procedure relies on testing all possible causal graphs for *consistency* with the observed data set by using the assumption of statistical independence between the noise of a variable and its parents. Consistency is tested by performing a non-linear regression of each variable X_i on its parents. Then, a non-parametric statistical independence test is performed to test the independence of the regression residuals and X_i. If the residuals are found dependent on any X_i, the causal model is rejected. Only models that pass all of the independence tests are considered consistent with the data. As the approach returns all of the consistent models, graphs that contain subgraphs consistent with the data are discarded from the results. In the case of linear f_i and Gaussian noise, non-linear ANM may not able to distinguish between orientations $A \to B$ and $A \leftarrow B$. Neither the non-linear regression method nor the non-parametric statistical independence test is fixed, but any appropriate methods such as Gaussian process regression or kernel conditional independence tests can be employed [40].

Due to the requirement of iterating over all of the possible causal models, the non-linear ANM can be applied only on low-dimensional data. However, with a large data set the algorithm can provide a useful method for checking a causal model. Consider a causal model that is obtained with some faster algorithm, by elicitation from a domain expert, or both. Non-linear ANM can help test the model for consistency with the data. Depending on the result, the model is either confirmed or discarded.

If the available data is known to contain linear relationships and the distributions of the noise terms are non-Gaussian, LiNGAM has been shown to produce good results. On the other hand, non-linear ANM provides a more general method than LiNGAM

when such assumptions do not hold or are not known to hold, although its computational complexity limits its use to low-dimensional data. Other methods based on structural equation models exist with different sets of assumptions regarding the data-generating process. For example, the post-nonlinear model (PNL) relaxes the assumption of additivity of the noise by defining variables as $X_i := f_{i,2}(f_{i,1}(\mathrm{Pa}(X_i)) + E_i)$ [116].

4. Model Selection

On a general level, the problem our approach as well as other causal discovery methods attempt to address is that of *model selection*. Model selection refers to studying the performance of a set of models with varying complexities in order to select the best one according to some pre-defined criterion, often predictive accuracy [e.g., 33, 42]. Once the best model with respect to the chosen criterion has been selected, its performance in terms of generalisability to previously unseen data is evaluated, which is called *model assessment* [33]. Model selection forms a complex task which has received attention, for example, in definition of the Bayesian workflow [28], which is discussed in Section 8.

In a causal setting, the task is to identify which one of a finite set of alternative causal structures produces the best fit for the observed data. Prior knowledge is not usually considered in model selection because the focus of the procedure is to identify which model fits the data best and the prior distribution contributes towards parameter choice rather than predictive accuracy [27]. Besides evaluating how well models fit to the data, further criteria for estimating and comparing their performance include detecting whether the model has overfit or underfit the data and whether distributional changes occur in the process that generated the data. Methods for comparisons among various models, for evaluating their goodness-of-fit to the data, and for detecting problems in them are thus needed to find a model with a good performance according to chosen criteria. In Subsection 4.1, we introduce the problem of balancing a model's bias and variance. We address model scoring in Subsection 4.2 and discuss cross-validation in Subsection 4.3.

4.1 Bias-variance Trade-off

The sample size of available data in model building is often limited which problem is further exacerbated by the need to leave out some of the data from the training set to validate and test the built models. With insufficient data to train the models, high model complexity, or training the models for longer than necessary can lead to *overfitting* [e.g., 29, 33, 35]. Overfitting happens when the model begins to fit the noise in the training data instead of capturing the principal trends. On the other hand,

underfitting occurs when the fitting is terminated before the model has learnt relevant patterns from the data or when the model is too simple to describe them [33].

Model selection as well as model assessment is often performed by computing the model's prediction error that is obtained from applying a *loss function* $\mathcal{L}$ to data that the model was not trained on. Prediction error over a validation sample that is independent of the training sample can be referred to as the *test error* or the *generalisation error* [33]. When the training data set is fixed, the squared prediction error of a regression model can be broken down into three components: *bias*, *variance*, and *irreducible error* [29, 33]. Same general idea applies outside the regression setting and squared error. Both underfitting and overfitting relate to the trade-off between the bias and variance of a model in model selection. The irreducible error is sometimes called the *Bayes' error* [42].

Assume a model of the form $Y = f(X) + \epsilon$, where X represents the independent data and Y the dependent variables, f is some deterministic function, and ϵ denotes the stochastic noise in Y with zero mean and a variance of σ. The noise ϵ is further assumed independent of both Y and X. Given a training data set D, we can obtain an estimate of the function f, denoted by $\hat{f}$. Now, the decomposition of the prediction error with squared error loss is given by

$$
\begin{aligned}
& E[\mathcal{L}(Y, \hat{f}(X))] && (4.1)\\
&= E\left[(f(X) + \epsilon - \hat{f}(X))^2\right] && (4.2)\\
&= E\left[\left(f(X) - E[\hat{f}(X)] + \epsilon + E[\hat{f}(X)] - \hat{f}(X)\right)^2\right] && (4.3)\\
&= E\left[\left(f(X) - E[\hat{f}(X)]\right)^2\right] + E\left[\epsilon^2\right] + E\left[\left(E[\hat{f}(X)] - \hat{f}(X)\right)^2\right] && (4.4)\\
&= \left(f(X) - E[\hat{f}(X)]\right)^2 + E\left[\epsilon^2\right] + E\left[\left(E[\hat{f}(X)] - \hat{f}(X)\right)^2\right] && (4.5)\\
&= \text{Bias}^2\left(\hat{f}(X)\right) + \sigma + \text{Var}\left(\hat{f}(X)\right), && (4.6)
\end{aligned}
$$

The result holds because the expectation of the difference between a random variable and its expected value is always 0, the expected value of ϵ is 0, and ϵ is assumed independent of Y and X. Due to f being deterministic, $\left(f(X) - E[\hat{f}(X)]\right)^2$ is constant and can therefore be moved outside the expectation.

The decomposition of a model's prediction error helps understand the trade-off that governs model selection. An example of such a decomposition is shown in Figure 4.1. The irreducible error, σ, defines the lower limit of the prediction error as the name suggests [33, 42]. No model can achieve a lower error rate than the variance in the data-generating process. In a classification setting, an optimal model that obtains the minimal error is called the *Bayes' classifier* [42, 49]. Bias of a model measures the distance between an optimal model, the expected value of Y which is $f(X)$, and

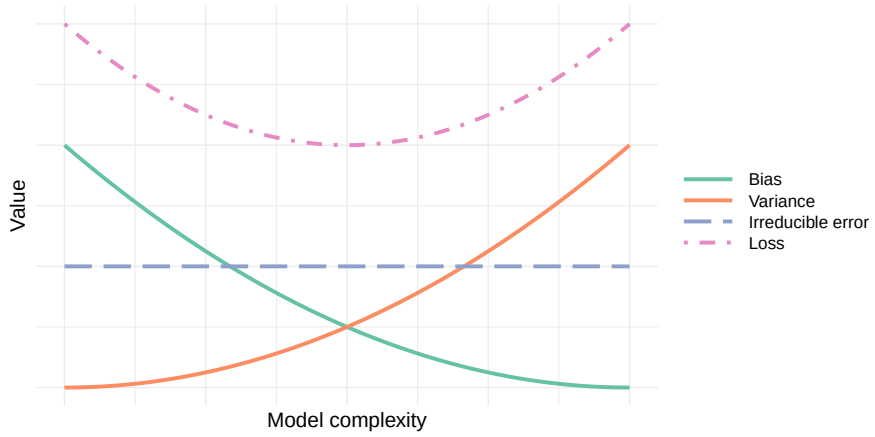
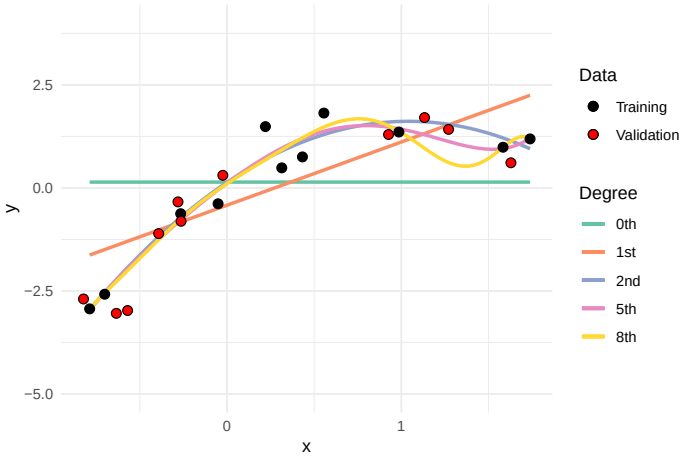


Figure 4.1: An example of a bias-variance decomposition. As the model complexity increases, the bias decreases and the variance increases. Irreducible error stays constant and loss is computed as a sum of the three.

the approximation [29]. A high bias indicates the model does not fit the data well and therefore produces poor predictions. Finally, the variance of a model estimates the model's dependence on the training data [29]. If approximations of a model were generated with different training sets, the variance expresses the expected variation in the estimations. High variance thus suggests a strong dependency between the estimated model and the training set, again leading to lower performance in predictions on new data.

Models that are too simple to estimate the data-generating process accurately tend to have high bias and low variance [29, 33]. On the other hand, complex models are able to produce accurate *in-sample* predictions, that is, predictions for values included in the training data set, but may not generalise well. Such models often have high variance but low bias. Underfitting can be detected by measuring a model's bias [33] and, conversely, high variance can be caused by the model overfitting the training data [29, 33]. Balancing these two opposite goals in finding a good model for a given process is referred to as the bias-variance trade-off. It is important to check models for each extreme, as avoidance of either high variance or low bias alone can lead to an increase in the other measure [83].

A common method for detecting underfitting and overfitting is validation [e.g., 33, 35, 42]. Generally, part of the available data, referred to as the test set or test data, is set aside from the training data for testing purposes and this data is not used for training at all. A model trained on the training data can be validated against another separate data set, often referred to as the validation set. The validation set is ignored during the model training, similarly to the test set, but can be used to compare a number of models to perform model selection. On the other hand, the test



Degree	Train	Validation
0	2.42	3.80
1	1.38	1.31
2	0.31	0.31
3	0.30	0.27
4	0.24	0.33
5	0.23	0.23
6	0.23	0.28
7	0.18	103.20
8	0.18	62.92

Figure 4.2: On the left, training and validation data plotted together with regression lines for some example models. On the right, a table of training and validation errors for the tested models with regression functions of varying polynomial degrees.

data is used only for assessing the performance of the final model to evaluate how well the model performs on previously unseen data. Essentially, the error computed on the validation set is an estimate of the test error and thus provides an approximation of the generalisation performance for model selection. The same data set cannot be part of both validation and model assessment because the validation data is used multiple times during the whole training process and the final model has therefore learnt to predict the validation data well. Testing the final model's performance on the validation data would provide a biased estimate of the true performance on unseen data.

As a general rule, the training error reduces with additional learning and more complex models due to higher flexibility allowing the model to fit the training data more precisely [e.g., 33, 42]. The validation error does not have the same property because the model is trained on a separate set and hence it helps determine the optimal complexity or amount of training for the training data. At first, the validation error decreases with model complexity until at some point it begins increasing. After the minimum of the validation error has been reached, additional model complexity leads to overfitting the training data. Conversely, before reaching the minimum validation error, the model can be deemed to underfit the data. The same interpretations apply for the amount of training executed as for the complexity. With just one validation data set the minimum is approximate but it still serves as a sanity check against overfitting and underfitting.

An example of model selection with a separate validation data set is shown in Figure 4.2. The true model is defined as a fifth degree polynomial with Gaussian noise. As expected, the training error decreases monotonically whereas the validation error first decreases then increases after it reaches the minimum value of 0.23. In this

example, the minimum validation error is obtained with the model complexity matching the true model which, however, may not be always the case with a limited validation set. Simple models fitted to polynomials of low degree such as zero or one result in large errors on both the training and the validation set. Large errors indicate a poor fit to the data or, in this case, the models having high bias. As the models get more flexible, the validation error explodes when the degree of the polynomial is increased to seven or eight, which serves as a sign of overfitting and the model capturing the noise in the data rather than the relevant patterns. The same results can be seen by studying the plotted regression lines in the figure but visually inspecting the models becomes impractical with higher dimensions.

Real-world data can contain extreme data points for which the predicted values according to a model differ much from the true values. Such data points are called *outliers* [e.g., 42]. They can be the result of measurement errors, execution errors [1], or a missing predictor indicating problems in the chosen model [42]. Execution error refers to how the data was sampled. If the data set consisted of air temperature measured during winter, a measurement from another population, such as the air temperature of some day in June, would appear as an outlier. Sometimes identifying why some samples have a different distribution than most of the data is not straight-forward. An example of such a situation is found in modeling air temperature over a number of years when a few of the samples are measurements from a year with extreme weather conditions.

If the training data contains outliers, the model fit can be skewed from the true trend resulting in a biased model that produces poor predictions for out-of-sample data [1]. Validation offers a simple means to assess the model fit, also applying to situations with outliers in the training set. Outliers in the validation set, on the other hand, can result in problems in model selection. For example, if the validation set contains a relatively high number of outliers for the process of interest, a model that captures the outliers' behaviour best could be selected instead of a model that fits the main trends of the process.

In a data set with *autocorrelation*, that is, data with dependent samples such as a time series, an outlier can occur as a single data point, a consecutive group, a period of data [22]. Cross-calidation can help detect groups of outliers in time series data, which is discussed further in Subsection 4.3, through identification of contiguous blocks of data that behave differently from the rest. If large errors are found on one of the validation sets only, it is possible it contains a group of outliers that affect the results, although the interpretation is sensitive to the context and further analysis of the data should be performed.

When building a model for some data-generating process, one basic assumption

is that all of the data is sampled from the same distribution and the relationships between variables stay constant. In real-world data, the assumption may not hold, as the relationship between the independent and dependent variables can change over time. The term *concept drift* refers to either abrupt or gradually forming changes in the distribution of the data or in the relationships between variables [23, 85]. Variables relate to each other through the conditional distribution of the dependent variable Y given the independent variables $\mathbf{X}$. *Virtual concept drift* refers to changes in the data distribution $p(\mathbf{X})$ and *real concept drift* to changes in the conditional distribution $p(Y \mid \mathbf{X})$, the relation between the dependent and independent variables [23]. An example of a gradual concept drift can be found in the decrease in battery life of a computer as a function of charging time over years. The change in battery life after replacing the old battery with a new one would be a rapid change in the process. In the case of gradual concept drift, observations located close together temporally can be assumed to be drawn from the same distributions whereas distant observations suffer from larger distributional differences. Concept drift is caused by changes in *hidden context* [113]. Hidden context of a process includes both interventions on the system, such as replacing an old battery, and missing explanatory variables such as those that cause a decrease in the performance of a battery over its lifetime.

In the context of models that are used for prediction, the occurrence of concept drift can render the trained model obsolete unless it is updated when the rapid change happens or when the gradual change has affected the results noticeably [23]. On the other hand, if models are used for understanding the underlying processes instead of predicting unobserved values, adapting the models to the changes may not be necessary. Detection of concept drift still carries relevance as one static model cannot describe the full data set well but separate models may be needed. Similarly to detecting outliers, cross-validation with contiguous blocks of data can help detect concept drift, through the inspection of model performance when the separate folds are used for validation. If models trained on data excluding a validation set perform poorly in the validation, concept drift can explain the discrepancy. However, the same result can indicate overfitting, misspecification of the model, or outliers, thus care must be taken in the interpretation.

4.2 Model Scoring

Even if producing predictions for a given process is not the primary goal of building a model, they can help estimate how well the model fits the data as well as detect possible overfitting and concept drift as discussed in the previous Subsection 4.1. Both model selection and model assessment can be performed based on measures of predictive

accuracy, which are referred to as *scoring rules* [27]. Scoring rules enable comparisons among a number of models as well as assessing the performance of a chosen model in terms of fitting the data. Common scores for measuring a model's fit to the data include the log-likelihood of a hold-out data set in probabilistic prediction or mean squared error (MSE) when point predictions are obtained [27]. In fact, if the model is Gaussian with a constant variance σ, the two measures are proportional.

$$\log p(y \mid \theta) = \sum_{n=1}^{N} \log \mathcal{N}(y_n \mid \mu(x_n), \sigma) \tag{4.7}$$

$$\propto -\frac{1}{2} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2, \tag{4.8}$$

where $\mu(x_i) = \hat{y}_i$ is the prediction output by the model for a new data point x_i. The result is the MSE of the model scaled by $-\frac{N}{2}$.

Although log-likelihood provides a good estimate of a chosen model to the data, its interpretation is not intuitive. On the other hand, the *coefficient of determination*, or R^2, provides an interpretable and common measure for evaluating model fit [42]. R^2 for a linear model represents an estimate of the proportion of variance in the dependent variable explained by the model, the true population value of which is denoted by ρ^2. It is computed with the ratio between sample estimates of variance under the chosen model and a *null model* that contains no predictors. Sample estimates of variance are computed using residuals from fitting the two models to the training data.

$$R^2 = 1 - \frac{\mathrm{Var}_M}{\mathrm{Var}_0} \tag{4.9}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N} (y_n - \hat{y}_n)^2}{N} \div \frac{\sum_{n=1}^{N} (y_n - \bar{y})^2}{N} \tag{4.10}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N} (y_n - \hat{y}_n)^2}{\sum_{n=1}^{N} (y_n - \bar{y})^2}, \tag{4.11}$$

where $\bar{y}$ is the mean of the observed dependent variables, representing the null model with zero predictors. Although R^2 often is defined only for measuring the predictive power of a model within a training set, an out-of-sample R^2 has been used with an equivalent definition [97, 9]. Out-of-sample R^2 can be evaluated by computing the sum of squared errors for the chosen model and the null model on a validation or test set, and subtracting their ratio from one. Similarly to the MSE, R^2 is proportional to the log-likelihood of a given data set when Gaussian distribution with a constant variance is assumed, as the denominator in Formula 4.11 is constant with regard to the data set.

The traditional definition of R^2 does not provide a good estimator for the effect of adding a predictor to the model as any new predictor always raises its value. This

behaviour is caused by biased estimators of variance in the formula. One simple solution to the problem is an *adjusted* R^2, denoted from here on by R^2_a, which is computed with unbiased estimators of variance [42]:

$$R^2_a = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2} \frac{N-1}{N-p-1} \tag{4.12}$$

$$R^2_a = 1 - (1 - R^2)\frac{N-1}{N-p-1}, \tag{4.13}$$

where N and p denote the sample size and number of predictors, respectively. With the ratio of the degrees of freedom of the chosen model and the null model, the variance estimators are made unbiased and the value increases when new predictors are introduced only when the increase in variance explained exceeds what is statistically expected. From Formula 4.13, it can be seen that the adjusted R^2 has a value lower than or equal to the unadjusted R^2.

The true proportion of variance explained by a model, ρ^2, falls in the range $[0, 1]$. When R^2 is estimated on the training data that the model is fitted on, the same condition applies if the model has actually been fitted to the data instead of selecting a model randomly or by some other criteria not based on the data. Computing the out-of-sample R^2_a for a validation set, on the other hand, can produce negative values if the model fits the data worse than a null model that always predicts the mean of the data set [97]. Even though a negative R^2 value reduces the interpretability of the metric, it serves to highlight poor generalisability and can help detect when the training and validation sets cannot be treated as samples from the same distribution.

As shown above, R^2 and, by extension, R^2_a can be computed on either training or validation data and under the assumptions of Gaussianity and constant variance they are proportional to the log-likelihood of the model. One approach to generalise the coefficient of determination to a Bayesian network is provided by its relationship to the log-likelihood in the case of a linear model with one dependent variable. The log-likelihood of a Bayesian network G with I variables is computed as a sum of the factors implied by its structure:

$$\log p(X_1, \ldots, X_I) = \sum_{i=1}^{I} \log p(X_i \mid \mathrm{Pa}_G(X_i)) \tag{4.14}$$

Let M be a linear model with some fixed parameters and a graph structure G and let the variables of M have Gaussian noise distribution with constant variance. The log-likelihood of each variable given its parents under model M is thus proportional to the R^2_a computed for the same variable under model M,

$$\log p(X_i \mid \mathrm{Pa}(X_i)) \propto 1 - \frac{\sum_{n=1}^{N}(x_{in} - \hat{x}_{in})^2}{\sum_{n=1}^{N}(x_{in} - \overline{X}_i)^2} \frac{N-1}{N- \mid \mathrm{Pa}(X_i) \mid -1}, \tag{4.15}$$

where $\hat{x}_{in}$ is the prediction made under M for the nth sample: some function of the nth sample of the parent set $\mathrm{Pa}(X_i)$. From equations 4.14 and 4.15, it follows that the log-likelihood of the whole Bayesian network G is proportional to the sum of R_a^2 computed separately for each variable. Furthermore, the log-likelihood is proportional to the mean of the R_a^2 values, denoted by $\overline{R}_a^2$.

$$\log p(X_1, \ldots, X_I) \propto \frac{1}{I} \sum_{i=1}^{I} \left(1 - \frac{\sum_{n=1}^{N} (x_{in} - \hat{x}_{in})^2}{\sum_{n=1}^{N} (x_{in} - \overline{X}_i)^2} \frac{N-1}{N- \mid \mathrm{Pa}(X_i) \mid -1} \right) \tag{4.16}$$

The resulting value does not cover the full interval $[0, 1]$ for a training data set, as each Bayesian network contains at least one variable with no parents by definition. Its range is thus given by $[0, (I-1)/I]$ for a Bayesian network with I variables as at least for one variable $R_a^2 = 0$. However, by multiplying the value by $I/(I-1)$ it can be fixed to cover the same range $[0, 1]$ as the regular coefficient of determination, or $[-\infty, 1]$ when computed on a validation set. We get as a result the following metric:

$$\overline{R}_a^2 = \frac{1}{I-1} \sum_{i=1}^{I} \left(1 - \frac{\sum_{n=1}^{N} (x_{in} - \hat{x}_{in})^2}{\sum_{n=1}^{N} (x_{in} - \overline{X}_i)^2} \frac{N-1}{N- \mid \mathrm{Pa}(X_i) \mid -1} \right) \tag{4.17}$$

Because of the final scaling, $\overline{R}_a^2$ has the two-variable R_a^2 as a special case.

The advantage of using R^2 rather than the model's log-likelihood to measure goodness-of-fit stems from its easy interpretation as the proportion of variance explained. $\overline{R}_a^2$ is essentially the log-likelihood of a Bayesian network transformed to the same scale as the coefficient of determination. The value does not represent the proportion of variance explained, but the mean of variances explained by regressing each variable on its parents. Despite the difference in interpretation to that of R_a^2, $\overline{R}_a^2$ is a metric for probabilistic graphical models that provides a more understandable measure of comparison between two or more models than the often used log-likelihood. It scales in terms of percentage of explained variance whereas changes in the log-likelihood are not linked to an easily interpreted quantity.

4.3 Cross-validation

Two levels of generalisation can be distinguished for test error: *prediction error* (PE) and *expected value of prediction error* (EPE) [4]. PE measures how well a model trained on a specific training set generalises to unseen test data whereas EPE can be seen as a more general measure of a learning algorithm or a model structure. PE is computed as the expectation of the selected loss function $\mathcal{L}$, for example the MSE, over new samples from the same distribution P from which the training set D that the model is fitted on is sampled. EPE is defined as the expectation of the prediction error over training

data sets of a given size.

$$\mathrm{PE} = E[\mathcal{L}(Y, \hat{f}(X)) \mid D] \tag{4.18}$$

$$\mathrm{EPE} = E[\mathcal{L}(Y, \hat{f}(X))] = E[\mathrm{PE}], \tag{4.19}$$

where X and Y are *independent and identically distributed* (i.i.d.) samples from distribution P.

As we do not know the generating distribution, the exact PE and EPE cannot be computed [4]. Approximations $\widehat{\mathrm{PE}}$ and $\widehat{\mathrm{EPE}}$ for the two errors can be found, which introduces a new task of evaluating the uncertainty in the approximations. Common methods to evaluate $\widehat{\mathrm{PE}}$ and $\widehat{\mathrm{EPE}}$ include *cross-validation* (CV) and *bootstrapping* which are also used to estimate the uncertainty of the found approximations. For some estimated quantities used to measure goodness-of-fit, such as R^2, analytical solutions exist to estimate the confidence intervals [92], although the same solution is not trivially extended to cover the $\overline{R}_a^2$ measure discussed in the previous Subsection 4.2. Although it has been proven that no unbiased estimator of the variance of the k-fold CV error exists [4], an almost unbiased estimator of the error variance has been shown possible by using nested CV [105], defined later below.

Regular k-fold CV is implemented by first dividing the data used for training and validation randomly into k folds [e.g., 33, 96]. Any sequential or other correlation structures in the data set can be ignored when forming the blocks. Each fold is used in turn to validate a model trained on the remaining $k-1$ folds. The cross-validation error CV_e is an average over the loss computed for the validation data in each iteration [96].

$$\mathrm{CV}_e = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}(Y_{\in i}, \hat{f}_{-i}(X_{\in i})), \tag{4.20}$$

where $Y_{\in i}$ and $X_{\in i}$ denote the dependent and independent samples in the validation fold i. $\hat{f}_{-i}$ represents a model fitted to the training sample that contains all of the data not included in fold i. A special case of k-fold CV is the *leave-one-out cross-validation* (LOOCV) where k equals the sample size, thus validating the fitted model on just one data point in each iteration. One iteration of k-fold CV with $k = 4$ and one of LOOCV are visualised in Figures 4.3a and 4.3b, respectively.

The main benefit of applying cross-validation is efficient use of data which often is scarce. As the training set is partly different in each iteration, the CV error provides an estimate of the expected prediction error, EPE [4]. Two considerations should be taken into account when choosing the number of folds k [33]. A large number of folds produces less biased estimates but, on the other hand, leads to a high computational cost and the estimate can have large variance. CV with few folds requires less resources and results in small variance but high bias. On the other hand, it has been proven that

the size of the fold should disperse towards infinity at the same rate as the sample size grows and that LOOCV does not produce asymptotically optimal results [86].

One source of bias in the CV error estimation stems from making modelling choices outside a CV loop which can provide the model with an advantage leading to underestimations of the error [105]. For example, if feature selection is performed using both training and validation data, the validation data no longer serves as an unseen test data set as the training of the model includes information from all of the data. *Nested CV* has been proposed as a solution to this problem [105]. In the approach, all modelling choices are made inside a k-fold CV loop with the relevant training data. The training data is further split into k' folds used for another CV loop within the current iteration.

A further complication in measuring the goodness-of-fit of a model is introduced by having correlated samples, such as those in a time series. We cannot assume the data points are i.i.d. since autocorrelation introduces a temporal dependency between the samples. The risk of overfitting increases when data that is generated by a dynamic process is sampled over a limited time period. For example, if we have data on weather conditions for June in a single year, training a model on that data set results in a model that has specialised on data with a similar distribution. In such a case, the model represents the available data set too faithfully and does not provide a general understanding of the data-generating process in other contexts. With yearly variance in weather conditions, the same model may not perform equally well on preceding or subsequent years. The problem is how to measure generalisability of a model given only a limited set of samples, which may not be independent samples from the same distribution. One approach is to use CV by splitting the data into chunks of equal size that are sufficiently large to account for the autocorrelation [5]. Unlike in traditional CV on i.i.d. data, the splitting cannot be performed fully at random since data points located temporally near each other are correlated.

A base case of validation in the context of time series data consists of splitting the series into two data sets of which the first is used to train a model and the second to validate it [8, 98]. The approach is sometimes referred to as *fixed origin evaluation* [98] and an example of applying it on a temporal data set is shown in Figure 4.3c. A more general version of out-of-sample evaluation of time series data is leaving out a block of data from the end of the time series for evaluation and training the model on a number of preceding data points, which is known as *last block validation* [5]. Last block validation, of which fixed origin evaluation is a special case, only refers to how the data is split and can include splitting the data multiple times. *Rolling-origin evaluation* either with *recalibration* or *updates* and *rolling-window evaluation* together with the basic split into two are examples of last block validation [98]. In rolling-origin-

Figure 4.3: Various approaches to prediction error estimation. Columns represent one iteration of validation on temporally ordered data with first sample at the bottom and the last at the top. (a) k-fold CV, (b) Leave-one-out CV, (c) Fixed origin evaluation, (d) Rolling origin evaluation, (e) Rolling window evaluation, (f) h-block CV, (g) Modified CV, (h) hv-block CV, (i) Blocked CV

recalibration evaluation, the data is split into a number of chunks and each chunk is in turn used to validate the model that is trained on all of the preceding chunks, as displayed in Figure 4.3d. Rolling-origin-update evaluation uses a similar approach but rather than training a new model in each iteration, the same model is used with updated inputs. Instead of all of the preceding values, training a model on a set number of preceding chunks to and then validating the trained model on the block following the training data is known as rolling-window evaluation, shown in Figure 4.3e.

Regular k-fold CV with random train-validation split ignoring the temporal order of train and validation data points has been applied to *autoregressive* models where each observation is modelled as a function of a set number of preceding observations [6]. The assumptions required to render the approach theoretically valid include having a *stationary* autoregressive model whose sample mean converges to its expected value in the limit of infinite data. The joint probability distribution of a strictly stationary model does not change over time [31]. In the presence of concept drift, for example, the validity of regular CV is not guaranteed. In practice, the theoretical guarantees may not be needed although stationarity and approximate independence of observations

with time lags over a specified threshold are commonly assumed [5].

Forming validation sets with mutually independent observations can be done by *h-block cross-validation*, shown in Figure 4.3f, which generalises LOOCV to dependent data sets [8]. For each validation fold, h observations that precede and succeed the observation used for validation are left out from the training set. The approach guarantees the training data are approximately independent of the validation data as long as h, the number of samples to remove, is chosen appropriately. Although h-block CV enables the use of any samples in a validation fold, the training set size is reduced significantly which reduces data efficiency. Similarly to LOOCV, h-block is inconsistent and does not produce asymptotically optimal results when used for model selection [73, 86]. A modification of this approach termed *modified cross-validation* (MCV) allows folds with size larger than one with each fold consisting of a random subsample of the original data, essentially generalising regular CV which corresponds to having $h = 0$ [2, 5]. An example iteration of MCV is displayed in Figure 4.3g.

In order to overcome the inefficiency regarding data use in h-block CV while still providing a valid evaluation method for time series data, *hv-block cross-validation* has been proposed as a generalisation of LOOCV and h-block CV [73]. Parameter v is used to control the size of the validation fold and h to ensure approximate independence. Each data sample z_i with index $i \in \{v, \ldots, n - v\}$, where n is the sample size, is in turn chosen as the centre point for the validation fold. The v samples preceding and succeeding z_i are included in the validation fold, resulting in a validation set of size $2v + 1$. All of the remaining data except for the h samples on either side of the validation fold are used for training to ensure approximate independence of the training and validation data as in h-block CV. The training set size in each iteration is $n - 2v - 2h - 1$ or at most $n - 2v - h - 1$, reaching the maximum if there are no samples on one side of the validation block. An example of hv-block CV is shown in Figure 4.3h. A related approach, *blocked CV*, uses blocks of mutually exclusive and continuous data as CV folds without excluding any additional data point from the training data, which equals hv-block CV with $h = 0$, as shown in Figure 4.3i [5].

Generally, the samples included in the validation or test set have to succeed the training set, whether there are gaps between the two data sets, in order to simulate real life conditions as the goal often is to use past data to predict the behaviour of future data [98]. A number of order-ignorant evaluation methods introduced above have been designed specifically for time series data to allow more efficient use of the available data than achieved with order-faithful methods. Even though temporal ordering should be taken into consideration, an empirical study found little difference in error estimation between various cross-validation methods [5]. The study compared regular CV, blocked CV, modified CV, last block validation, and a variation on last block validation where

validation is performed on the second instead of the last block to study the effect of temporal ordering on the error estimates. Each method was applied on a number of both simulated and real data sets. Results show that using only one split of the data, as in last and second block evaluation, produces higher variance in the error estimates, but no significant difference was detected between the CV approaches in terms of producing too high or too low estimates. Although the error estimates were found equally robust with the three cross-validation methods, modified CV uses data less efficiently than regular or blocked CV. The researchers conclude with a recommendation to use blocked CV although regular CV performed in the study equally well. All of the tested time series data sets are stationary. As a consequence, the results as well as recommendations made may not be applicable to non-stationary data.

The cross-validation methods introduced above have been proven valid only for stationary time series. Concept drift as a source of non-stationarity can be detected, however, with cross-validation if the validation folds are chosen not at random but as contiguous blocks of data, as described in Subsection 4.1. Either blocked or *hv*-block CV allows detecting both concept drift and outlier periods of data. By inspecting the validation errors separately for each fold, we can identify blocks that are more difficult to predict than others when training the model on the remaining data. Gradual concept drift is harder to detect than abrupt changes in the data distribution. By performing CV with different parameters, differences between training and validation scores for certain blocks can imply either outlier regions or concept drift. Divergence of the validation and training scores can indicate overfitting and such cases should be investigated further to determine the cause of the divergence. The inspection, again, can be done by reapplying CV on a subset of the data, additional data, or with different number of folds. If the scores are found to converge when leaving out certain data folds, the left out data may contain outliers that skew the results. Concept drift can be the cause if the results vary depending on whether a subset of data from the beginning, middle, or the end of the available observation period was used. Additional data can help avoid overfitting.

The cross-validation methods discussed are defined for contiguous time series with regular intervals and gaps. Non-contiguous time series data does not necessarily introduce problems as gaps longer than the autocorrelation period of the data actually ensure at least approximate independence of the samples on either side of the gaps. These gaps thus allow the use of methods such as blocked, *h*-block, or *hv*-block CV where some of the left-out data can be replaced with actual missing values, raising the ratio of observed data used for training. Especially with *hv*-block CV the blocks can be chosen to border with existing gaps in the data and additional data can then be left out only when there is no gap before or after a validation block.

5. Applied Causal Modelling

Causal modelling has received much attention in the Earth system sciences recently, such as in a review on commonly applied methods [79]. Excepting the fast hill-climbing algorithm [24], an example can be found of a practical application in the field for each of the algorithms discussed in Section 3. Due to difficulties in finding causal relations from purely observational data, some focus has been directed towards developing methods for incorporating knowledge from domain experts into causal analysis. Below we provide a short review of work related to both causal modelling in Earth system sciences in Subsection 5.1 and causal modelling for domain experts in general in Subsection 5.2.

5.1 Causal Modelling in Earth System Sciences

Granger causality [30] has gained popularity in practical research [e.g., 45, 91], possibly due to its clear definition and simple application. Air temperature in the Southern hemisphere has been found to Granger-cause air temperature in the Northern hemisphere [45]. Further analysis shows the causal ordering between the two variables probably results from human activity, providing evidence for humans likely contributing towards global trends in air temperature. In another study, carbon dioxide has been determined to be a main cause for the global surface temperature when compared with solar and volcanic activity as possible drivers [91]. The relationship is tested with an extension on Granger causality designed to detect long-term causal relations from time series data. However, the usage of Granger causality has been criticised for investigating the effect of human activity on global temperatures. The approach is argued to be inappropriate due to long lags between the cause and detectable effects [100]. Although an extension has been proposed to take into account long-term causal connections [48], the extended method requires making additional assumptions including Gaussianity of the variables which may not always be met. The possibility of latent variables is mentioned as another limiting factor in the interpretation of results [48].

One of the earliest studies on the PC algorithm [93] in climate science applied it to identify the causal relations between the gravity-adjusted heights from sea-level at different locations in a *climate network* [18]. The climate network is defined a grid

of N locations on Earth with temporal dimension taken into account by including 15 temporally sequential observations for each location in the network. Causal analysis is thus performed on a climate network of size $N \times 15$. Climate change has been studied with the same method but applying it on data sets describing different time periods obtained from a simulation climate model and comparing the results with each other [17]. The study focuses on describing the change in climate rather than performing causal analysis to form claims about the underlying causes behind the phenomenon.

Most methods for causal analysis rely on the assumption of causal sufficiency, that is, including in the model observations of all of the common causes of two or more model variables. To overcome this limitation, the suitability of the fast causal inference (FCI) algorithm [94], which allows the presence of latent variables, has been tested on climate data [81]. The algorithm is applied to detect causal relationships between Arctic temperatures and the jet streams. Results are compared with those from another study [82] that applies PC to the same problem with the same data. Although the approach is found promising, its limitations include lack of robustness and reliability [81]. Furthermore, the high computational cost of FCI has been proposed as one reason for the lack of practical applications for it [19].

The performance of a number of causal structure discovery (CSD) algorithms has been studied by applying them to detect the causal structure between an *urban heat island* and urban rainfall in the summer [57]. Urban heat island is defined to reference an urban area with a higher air temperature than surrounding rural areas. The six algorithms compared comprise PC, its two variants PC-Stable [12] and conservative PC [76], linear non-Gaussian additive models (LiNGAM) [87], and two variants of greedy equivalence search (GES) [11, 60]: fast greedy search [75] and independent multiple-sample GES [74]. Running PC, its variants, and LiNGAM on the available observational data produce equal causal structures, even though PC assumes Gaussianity and LiNGAM, conversely, assumes non-Gaussianity. All in all, the researchers find the results promising with an additional recommendation of including prior knowledge in the analysis for robustness.

As another example of an algorithm based on structural equation models in addition to LiNGAM, the performance of non-linear additive noise models [40] has been studied on multiple bivariate problems in the field of geoscience and remote sensing [71]. The method is applied on a set of 28 bivariate causal inference problems in geoscience. Ground truth models used for evaluation are formed based on expert knowledge or common sense. In addition, the approach is tested on a set of 182 causal problems based on simulated data sets with existing ground truth models. Again, the results are found encouraging as causal relationships are correctly detected with orientation taken

into account better than expected by chance alone. The researchers add as a caveat that the method is unsuitable in situations where the assumption of additive noise is not met.

Specifically designed for causal analysis of time series data, PCMCI [78, 80] provides a suitable method for causal discovery in Earth system sciences by combining the PC algorithm with momentary conditional independence. The approach has been applied to data on air pressure at sea level generated from model simulations as well as observed data [63]. Causal structures built from the simulated data are compared with the structure found for the observed data in order to evaluate the accuracy of the models and detect commonalities in the models' development. The proposed causal model evaluation provides a useful framework to evaluate climate models. Possibility of concept drift in future observations and missing variables are mentioned as limiting factors.

The PCMCI algorithm has also been used to identify the causal structure of carbon dioxide flow [51]. Three data sets collected from three stations located around the same area are studied to detect causal links in the flow process. Split by month and filtered for daytime observations to account for seasonal variance, the data are analysed with PCMCI to determine links that are found in one, two, or all of the data sets. Although consistency is observed in the discovered edges, the results have high false positive rate when assumptions such as linear relations and stationarity are not met. A majority of the found causal connections are found to be contemporaneous which prevents the algorithm from distinguishing causes from effects. As the algorithm does not take into account common causes or mediators with zero lag, the found structure can include spurious links. Nevertheless, the method is found to produce robust results with better interpretability than methods purely based on analysing correlations in the data.

5.2 Causal Modelling for Domain Experts

Proposals have been made for using prior knowledge together with CSD algorithms to detect causal relationships from observational data. A common approach is suggested as first eliciting relevant information from a domain expert and then incorporating that knowledge into the analysis [e.g., 60, 64, 110]. The analysis itself is performed by an expert in causal inference rather than the domain expert and the stages of knowledge elicitation and causal analysis may be executed iteratively. Such an iterative method for incorporating domain knowledge has been applied, for example, in medicine [21]. The same approach with only one iteration has been applied to sea-breeze prediction [46]. Knowledge elicited from experts can include information on correlations, *temporal or-*

derings, and the presence or absence of causal relations [64]. Temporal ordering for a model is defined by specifying variable groups that can be sorted by their relative temporal occurrence and the ordering may be provided partially for a subset of the model variables or for the full set of variables. The groups of variables used to define a temporal ordering are referred to as *temporal tiers* [64]. Another study suggests the use of a causal model's distance to a candidate model provided by a domain expert as a prior for CSD [37]. The TETRAD software package has been developed to facilitate the use of causal inference methods for domain experts directly [84]. The package allows for incorporation of background knowledge into CSD in the form of temporal tiers [54, 64, 84].

A recent paper presents a system, Outcome-Explorer, for building and exploring causal models interactively [39]. The motivation for the system stems from the need for explainable AI and it is targeted at both expert and non-expert users. Focus is placed on the interpretation of an existing causal model although the provided solution includes a module for causal model discovery. The user inputs a data set and then selects one CSD algorithm whose output is displayed to the user. They may then edit the selected model, at the same time evaluating the current model based on a number of metrics and value manipulations. With the emphasis on studying the relations in causal models, the model discovery performed with algorithms and expert interactions is not given an explicit mathematical formulation. The navigation in the space of causal models begins from a single graph although multiple algorithms are provided for finding the initial model. Outputs from different algorithms are not displayed simultaneously in a single frame for comparison.

The Visual Causality Analyst has been designed to facilitate the discovery and visualisation of causal structures from observational data sets with both continuos and categorical variables [111]. Analysis in the proposed system begins with applying an algorithm similar to PC on the data set to identify a possible causal structure. After the initial model has been found, the user may apply edits on the model by adding, reversing, or deleting edges. Linear and logistic regression are used to estimate the strength of the found causal relationships for continuous and categorical variables, respectively. Once a variable has been selected, the appropriate regression method is applied depending on the type of the variable. A number of metrics are computed from regressing the variable on all of its direct causes in the current model. The metrics include the p-values for regression coefficients and the R^2 value.

Stating Simpson's paradox as a motivator, the creators of the Visual Causality Analyst propose another interface, Causal Structure Investigator, that allows dividing the input data set and building separate causal models for each of the partitions [112]. As with the previous approach, the analysed data set can contain both continuous and

categorical variables, a similar, PC-based algorithm is used for CSD, and the model parameters are estimated with linear and logistic regression depending on the data types of the variables. The main contribution of the new framework is the manual or automatic division of data to enable building separate causal models for the partitions and comparing the found models. By selecting one or more features, the user may manually set ranges or values to generate data partitions. Alternatively, clustering algorithms such as k-means may be run on the data to find a specified number of clusters. Once causal models have been built for each of the data partitions, the models represented as adjacency matrices can be clustered with k-means to find groups of models with similar structure and models most representative of each cluster are identified for the user. If the data is divided into equal sized partitions, pooling of the causal links within model clusters can be perfomed by aggregating edges weighted by the BIC scores of the containing models.

SeqCausal has been developed as a system for the causal analysis of multiple event sequences and the visualisation of the found models [43]. The proposed approach is designed to address the incorporation of expert knowledge into causal analysis of data sets from different sources, for example, multiple patient records. Granger causality is used as the underlying method for iteratively detecting causal structure by incorporating the user's input into the CSD process. Through interactions, the user may impose constraints on the causal structure which are integrated into the objective that is optimised in each iteration. To enhance interpretability of the results especially when the data contains a large number of variables, any links can be viewed separately from the full causal structure. The separate visualisation is constructed to help the user determine the probability of the existence of a causal link by displaying the proportions of event sequences where the effect follows the cause, the effect is not preceded by the cause, and the effect does not follow the cause. Causal structures built for separate subsets of the event sequences can be compared with BIC as the metric for scoring models. The same score is used to track the effects that user modifications have on the goodness-of-fit of the causal models.

CausalMGM [26] provides a method for visualising results that are obtained by running a PC-based CSD algorithm on mixed categorical and continuous data. To mitigate the curse of dimensionality, the user may perform feature selection to choose a given number of variables most correlated with a target variable but whose correlation with each other is minimised. First, all of the variables excluding the target variable are sorted in descending order according to the correlation coefficient between them and the target. Second, the next variable according to the ordering is moved to a set of chosen variables if its correlation with any of the variables already in the set does not exceed a pre-defined threshold. The second step is repeated until the set of

chosen variables reaches the correct size or until no more features remain to choose from. Causal structures are found with a modified version of PC-Stable designed to allow for both continuous and categorical variables. Focus of the approach is placed on facilitating CSD for domain experts and visualisation of the results. Editing the found models is not possible within the proposed framework and interactions comprise annotating the resulting causal graph's edges and vertices.

Gathering information from domain experts is not trivial [25] and that information is prone to a multitude of biases [102]. In our approach, however, rather than eliciting prior probability distributions, we ask the expert to incorporate their beliefs regarding the conditional independence relations between the variables in the model. This task has been stated to be a simpler and more straight-forward one than probability elicitation [25]. Explicitly stating the assumptions brought into the model by the expert further alleviates the issues of uncertainty. If all of the assumptions that are made during the analysis are known, they can be scrutinised after a good model has been found and listing the assumptions enables replication of the obtained results. Furthermore, assumptions made during the model discovery can provide ideas about which experiments should be performed in order to fix the model.

6. Methods

We propose that interactive causal structure discovery (interactive CSD) should contain obtaining a selection of possible initial models, navigating in the space of causal models to incorporate expert knowledge, and employing validation to detect problems such as overfitting and concept drift. We do not aim to offer a definitive answer to how these modules should be formulated or implemented but present one alternative for manifesting the workflow. For example, we assume the user to be a rational Bayesian agent with a constant prior that they do not update during the navigation. The problem could be given a more complex formulation than the one we present.

In this section, we introduce one theoretical formulation of interactive CSD with an expert user and describe our practical implementation. The computational formulation of the problem and our proposed solution is discussed in Subsection 6.1. In Subsection 6.2, we introduce our implementation of the solution.

6.1 Formulation

Given a data set X, the task is to find a causal structure modelled as a graph that fits X and agrees with the expert's prior knowledge. We formulate the problem as Bayesian probabilistic modelling: X is assumed to be a sample from a probability distribution $p(X|\theta)$ where the parameters θ define a causal model over the variables in X. If we have a prior distribution $p(\theta)$ over all possible causal models, then Bayesian inference provides the posterior distribution $p(\theta|X)$, after observing the data set X. We propose a greedy optimisation of the posterior to find the maximum a posteriori (MAP) model or at least a model with a locally optimal probability. The user is assumed to be a rational Bayesian agent. The data set X is a $\mathbb{R}^{n\times p}$ matrix with the rows denoted as $(x_1, \ldots, x_n)$, each x_i a vector of p observations.

In our formulation, the parameters θ are split into two: (θ, β), where θ represents parameters about the structure of the causal graph as edge probabilities, and β represents the remaining model parameters, such as functional forms of parent-child relationships, regression coefficients, and noise distributions. Their joint distribution is then factorised as $p(\theta, \beta) = p(\beta|\theta)p(\theta)$ which allows us to separately specify a prior over

the structure and a prior over the model parameters given the structure. We assume that the data has been generated by a model with fixed but unknown "true" sets of parameters, θ_T and β_T, and that our data set has been sampled from the distribution $p(X \mid \theta_T, \beta_T)$.

For a given model structure θ, we define a *set of neighbours*, $N(\theta)$, as the set of models that can be reached by making one edit to the structure θ. An edit is a modification to the model structure by either adding, removing, or reversing an edge in the corresponding causal graph.

With the definition of conditional probability, the joint distribution of the full prior information can be written as $p(\theta, \beta) = p(\beta \mid \theta)p_U(\theta)$ where $p(\beta \mid \theta)$ is known by the computer and $p_U(\theta)$ by the user but typically not by the computer. We denote by $p_C(\theta)$ the prior distribution of θ assumed by the computer. The distributions $p_U(\theta)$ and $p_C(\theta)$ are not necessarily equal. We assume a causal discovery algorithm can find the maximum a posteriori model whose structure is θ_C with the computer's prior distribution $p_C(\theta)$. The computer's MAP solution is given by

$$\theta_C = \arg\max_{\theta} p(X \mid \theta)p_C(\theta) \tag{6.1}$$

where β has been integrated over, $p(X \mid \theta) = \int p(X \mid \theta, \beta)d\beta$, assuming the prior $p_C(\theta)$ for θ. In this formulation, the likelihood $p(X \mid \theta, \beta)$ as well as the priors of the computer are assumed given.

Our objective is to find the user's best solution θ_U, which is given as the MAP solution given the user's prior distribution:

$$\theta_U = \arg\max_{\theta} p(X \mid \theta)p_U(\theta) \tag{6.2}$$

However, as the computer does not have knowledge about $p_U(\theta)$, finding the solution θ_U is non-trivial. A greedy approach to estimating the user's best solution allows the user to perform local moves in the parameter space θ. For each model defined by θ the user can navigate to any state from its set of neighbours $N(\theta)$.

Beginning from the initial state obtained from applying a CSD algorithm, θ_C, in each step the user is assumed to greedily select a state that maximises the model's probability in the user's posterior. The states thus form a sequence $\theta_1, \theta_2, \ldots$ where $\theta_1 = \theta_C$. At iteration t, the next state is given by:

$$\theta_{t+1} = \arg\max_{\theta \in N(\theta_t)} p(X \mid \theta)p_U(\theta) \tag{6.3}$$

Once there are no more edits that would raise the posterior probability of the model given the user's prior, the user stops the exploration. With this process, θ_U or at least a local optimum of the user's posterior is found.

6.2 Implementation

To measure a model's fit to data, we use $\overline{R}_a^2$, adjusted coefficient of determination that is averaged over all of the variables in the model and scaled to cover the range $[0, 1]$, or $[-\infty, 1]$ for a validation set.

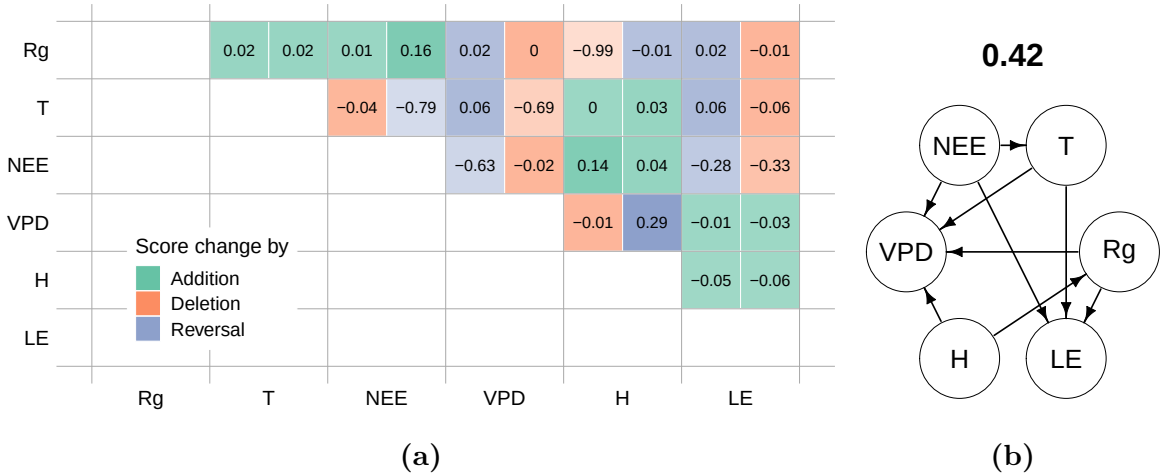
$$\overline{R}_a^2 = \frac{1}{I-1}\sum_{i=1}^{I}\left(1 - \frac{\sum_{n=1}^{N}(x_{in}-\hat{x}_{in})^2}{\sum_{n=1}^{N}(x_{in}-\overline{X}_i)^2}\frac{N-1}{N-\mid \mathrm{Pa}(X_i)\mid -1}\right) \tag{6.4}$$

As discussed in Subsection 4.2, $\overline{R}_a^2$ is proportional to model log-likelihood under the assumptions of Gaussianity and linearity. Although these assumptions may not hold in practice, we use $\overline{R}_a^2$ to estimate model log-likelihood for discovery of the approximate MAP solution as described in the previous Subsection 6.1. In practice, each variable is linearly regressed on its parents, adjusted coefficient of determination, R_a^2, is computed for that variable and, finally, the mean of the computed values multiplied by $I/(I-1)$ is returned as the model's score. The same method is used to estimate the score for both training and validation sets. Validation score is simply computed by training the regression models on the training data and then computing the $\overline{R}_a^2$ for the validation data.

Our approach is not restricted to a specific validation method but a suitable scheme may be chosen based on the characteristics of the analysed data. In the experiments, we use two different methods for validation depending on whether the data samples are independent of each other. Independent data are split into two equal-sized data sets one of which is used for training and the other for validation. Time series are validated with blocked CV [5]: the data are split into contiguous blocks each of which is used for validation in one iteration while the rest are used for training. Our main goal in the model selection is not to produce predictions based on the currently available data, but to find a good model which helps explain the underlying process that generated the data. Therefore, we can ignore the temporal ordering of the cross-validation folds when performing model validation. Final cross-validation score is computed as an average over the validation scores for the separate blocks.

The process of causal discovery begins with running a number of CSD algorithms on the data. Outputs from the algorithms are displayed to the expert user as circular graphs together with their respective validation scores. Variables are placed at same positions in the circular graphs to facilitate comparisons among the structures. The causal structure with the highest validation $\overline{R}_a^2$ is chosen as the initial model by default but the expert may choose any of the algorithm outputs to begin their navigation from.

In the experiments, we use a default set of algorithms and their parameters—PC with significance levels of 0.01 and 0.1, greedy equivalence search (GES), and linear



(a) **(b)**

Figure 6.1: Interface for expert interactions in CSD. Adjacency matrix (a) shows the effects of all possible edits on the validation score. Graph (b) shows the current causal structure and its validation score. Expert user may edit the current structure by clicking the cell of the adjacency matrix that corresponds to the edge being edited.

non-Gaussian models (LiNGAM)—although the algorithms could be chosen by the expert in practice. LiNGAM produces a single DAG as output but both PC and GES produce a completed partially directed acyclic graph (CPDAG) that uniquely represents a Markov equivalence class. To allow easy comparison among the results, we iterate through all of the DAGs in the equivalence classes output by PC and GES, and display to the user five highest-scoring DAGs from each class. If the data meets the assumptions of linearity and Gaussianity, all of the models that belong to a specific equivalence class are likelihood equivalent. In such cases, the five DAGs displayed to the user from each class are chosen randomly.

Current model is displayed both as an upper-triangular adjacency matrix by clicking on which edits to the model can be performed and as a circular graph, as shown in Figure 6.1. The graph is drawn with variables in the same positions as in the outputs of the CSD algorithms. At each step of the process, the effect of all possible edits on the validation score of the current model are computed as described above and shown to the expert in the adjacency matrix. Using this information together with their prior knowledge, the expert may then perform edits until satisfied with the current model. All of the edits that the expert performs are stored and shown to them throughout the process, allowing them to return to a previous state as well as state explicitly the assumptions included in the model discovery.

7. Experiments

In our approach, causal structure discovery (CSD) algorithms are used to provide initial models for the expert user to begin their analysis. The algorithms' outputs can act as "global" navigation points the expert may jump to instead of making local navigational moves with single edits. The algorithms included in the experiments comprise PC-Stable with two significance levels 0.1 and 0.01, greedy equivalence search (GES), and linear non-Gaussian additive models (LiNGAM). The main reasons for selecting these algorithms were to have a diverse group of algorithms based on differing approaches for which ready implementations exist. Our proposed procedure can be extended easily to other algorithms for which reason the set included at this stage is somewhat irrelevant. All of the selected algorithms assume causal sufficiency and linear causal relations.

Other algorithms with ready implementations that we considered were fast hill-climbing (FHC) [24] and fast causal inference (FCI) [94] but they were not included because the run time of FHC was too slow for the experiments given our resources and the results from FCI would not be comparable: FCI outputs a partial ancestral graph instead of a DAG or a Markov equivalence class as the other chosen algorithms. For all of the CSD algorithms, we used the implementations available in the R package `pcalg` [44]. The experiments were performed with version 3.6.3 of the R language [72] and the code is available online[†].

7.1 User Simulations

Synthetic data are created by generating random directed acyclic graphs parameterised by number of vertices and then sampling the graph with random edge weights for data sets of varying sizes. Each graph is generated with a sparsity of 0.3, meaning that with a probability of 0.3, each pair of variables is adjacent leading to graphs where approximately a little under one third of variable pairs are linked by an edge. Model variables are defined as an ordered sequence which is used as a topological ordering for the model. All of the edges are oriented according to the order, away from the

[†] https://www.dropbox.com/s/j88hd7xv9k5w41z/ICSD_kdd2021.zip?dl=0

first variable, to ensure acyclicity. The additive noise for each variable follows a zero-mean distribution which is randomly chosen from two options: either uniform (-0.1, 0.1) or Gaussian with a standard deviation of 0.1. The reason for including two types of noise distributions is to create data sets which almost follow assumptions made by the algorithms while still breaking some of them. All of the algorithms we use in the experiments assume linearity but additionally, PC and GES assume Gaussianity of noise and LiNGAM assumes non-Gaussianity. Selecting the noise distribution for each variable randomly from the two alternatives essentially creates data that does not conform fully to either set of assumptions.

A user is modelled with a parameter $k \in [1/3, 1/2]$ to represent their level of knowledge of the true states of all possible edges. Each pair of variables has three possible states for the connecting edge in terms of causal dependence: absent or present with an arrow in either direction. The prior for an edge thus consists of a discrete distribution over the three exclusive events. We assume the edges in a causal model to be independent and, therefore, can compute the prior distribution of the full graph as the product of the edge priors, or as a sum of the edge priors in log-space. As we know the true causal model, k determines the prior of the true state of an edge and the two remaining states have prior probabilities of $(1 - k)/2$ each.

In theory, k can take values in the range $[0, 1]$ as it represents probabilities. If $k = 1$, the user knows the true state of all edges absent from and present in the causal graph that generated the synthetic data with a probability of one, in which case the posterior is dominated fully by the prior and the model's fit to data is not taken into account. If $k = 1/3$, the user has no prior information about the causal structure and the posterior is dominated by the model likelihood as the prior is flat for all of the possible edges. We do not consider here the possibility of incorrect knowledge, so $k \geq 1/3$ always, and we found that values higher than $1/2$ do not produce interesting results as such high levels of knowledge lead to near-constant results.

In addition to studying the impact that the level of knowledge has on the results, we test the effect of partial knowledge where the user has information about two thirds of the variable pairs but no knowledge of the remaining pairs. This corresponds to using two values of k, one for the known parts of the graph and another, $1/3$, for the unknown parts. Edges with a flat prior regardless of the user's level of knowledge k are chosen randomly to cover approximately one third of all of the possible edges. Remaining edges are given priors with probability k for the true edge state as described above.

We use the structural Hamming distance (SHD) to compare causal models [16]. As discussed in Subsection 2.3, SHD between two graphs represents the number of edits required to transform one graph into the other, where an edit comprises either

adding, deleting, or reversing an edge. For each set of parameters (k, sample size, number of variables, proportion of unknown variable pairs), a hundred random graphs are generated to find meaningful distributions for the metrics used.

7.1.1 Experiment 1: Effect of Expert Knowledge

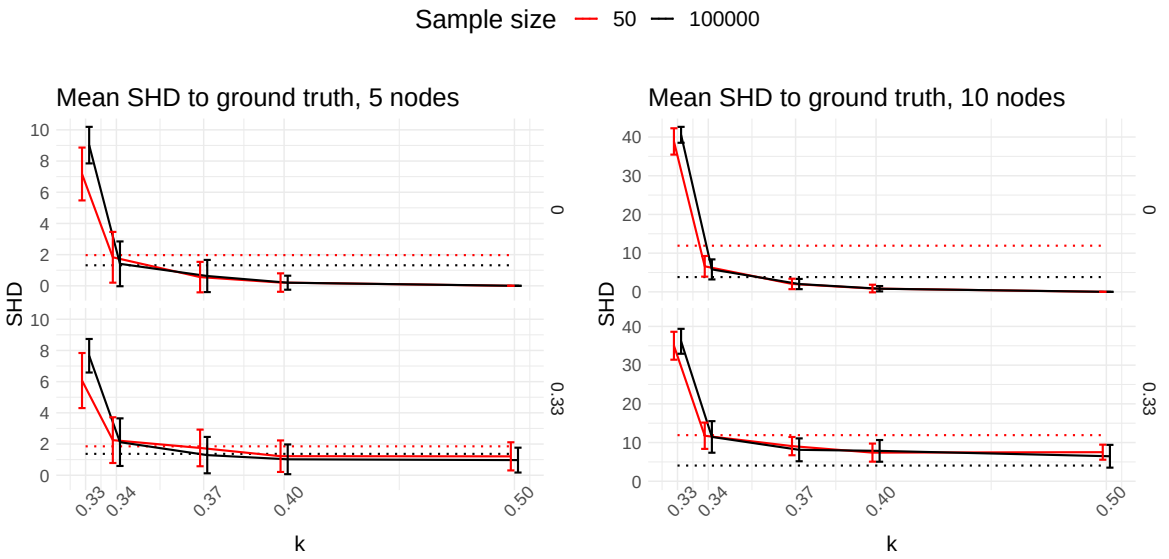
In Experiment 1, we examine how expert knowledge results in better models by simulating a user navigating in the model space. After running the default set of algorithms on a simulated data set, the highest scoring output is chosen as the initial model. The model is edited one step at a time, greedily selecting the neighbouring model with the highest user posterior.

For each model, the posterior is computed with the simulated user's prior, parameterised by k, combined with the model's approximate log-likelihood. Log-likelihood is estimated by linearly regressing each variable on its parent set in the model and computing the mean squared error of the fitted linear model on the validation data. The estimates for all of the model variables are subtracted from zero and the final value is divided by 2.

When the current model has the highest posterior over all of its neighbours, the navigation ends. Our approach is not designed to find some true model but a model with a locally maximal probability in the user's posterior. However, in our experiments, the user's prior corresponds to the true model with only the level of knowledge k varied and, thus, to evaluate the results, we use the SHD between the final models from simulations and the true model.

Figure 7.1 shows the results of Experiment 1. We see that with knowledge of all pairs of variables (upper figures), higher levels of user knowledge lead to models closer to the ground truth than the average initial model which is denoted by a dotted line. In contrast, greedily optimising the model score, which corresponds to using a flat user prior with $k = 1/3$, generally leads to worse models than the initial model in terms of SHD. This observation is explained by the true model not having necessarily the highest $\overline{R}_a^2$ which is proportional to the log-likelihood. For example, with 10 nodes and small sample size, the average score of the true model is 0.54 and the average for a model found with flat prior is 0.66. For 5 nodes and large samples, the corresponding values are 0.36 for the true model and 0.43 for the result of navigation with a flat prior. The results underline the need to rely on both the data-based score and expert knowledge to find good models.

Even when the user has no knowledge of the causal connections between a third of the variable pairs (bottom figures), user interaction improves the initial model when the data set is small or contains few variables. When the expert's knowledge has no



(a) Results for graphs with 5 nodes.

(b) Results for graphs with 10 nodes.

Figure 7.1: Experiment 1. Mean structural Hamming distances to the ground truth model. Error bars represent mean ± standard deviation, dotted lines the mean SHD of the initial model to ground truth. Above results with full knowledge, below one third of knowledge missing. Incorporating more expert knowledge leads to models closer to ground truth, especially with small sample sizes.

missing information (upper figures), the resulting models are closer to the true model than the initial model for $k \geq 0.34$ with small sample size and for $k \geq 0.37$ with large samples. This observation suggests that even low levels of user knowledge lead to better models.

As expected, expanding the sample size improves the performance of the CSD algorithms leading to better initial models as seen from the black dotted line being below the red one. However, high levels of knowledge still improve these initial models converging them towards the true model which, in these experiments, corresponds to the posterior model of the simulated user when no information is missing, the prior covers all pairs of variables. Adding more model variables (Figure 7.1b) expands the model space which negatively affects the initial model given by the CSD algorithms, leading to higher SHD values. Again, including user knowledge still improves the final result in all cases except in the case of large data and missing knowledge (bottom Figure 7.1b).

7.1.2 Experiment 2: Effect of Initial Model

In Experiment 2, six graphs are used as initial models for the navigation: an empty graph, the true graph, and the highest scoring model for each of the four default

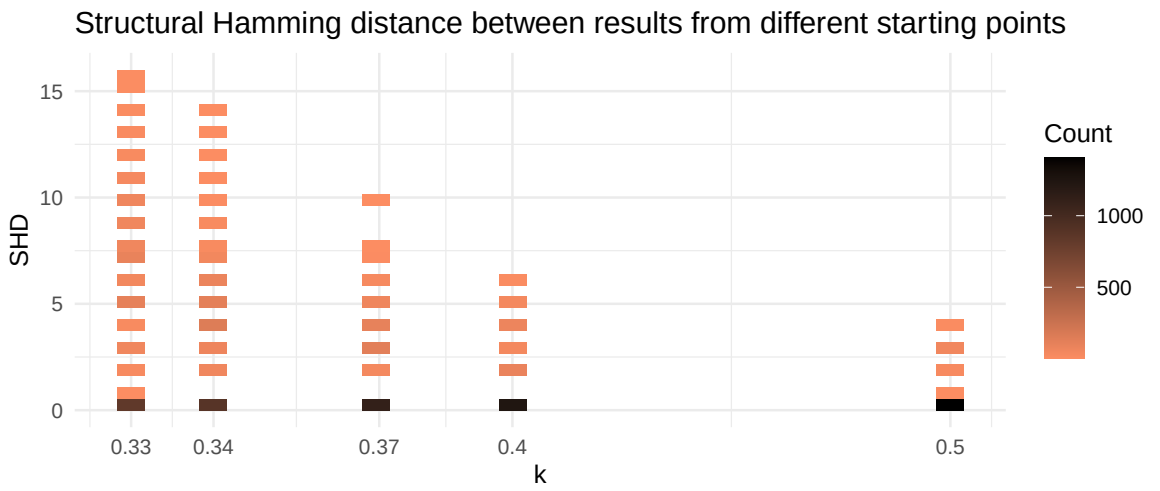


Figure 7.2: Experiment 2. Pairwise structural Hamming distances when running analysis on the same data beginning from different initial models. Variance in the distances shows the final model is affected by choice of initial model.

algorithms. Navigation is performed as in Experiment 1 but we use SHD to compare the resulting models with each other instead of with the ground truth. Comparisons among models found through navigation allow us to determine whether the initial model affects which model is found and, therefore, whether it is useful to have a number of different initial models to choose from.

Figure 7.2 shows the pairwise SHDs between final models when different initial models are used for the same data and level of knowledge. We used seven variables to generate the data sets each with a sample size of one thousand and set a flat prior to a third of the possible edges for all values of k. The resulting final models are mostly equal with majority of pairwise SHD values at zero but, with lower levels of knowledge, the results contain more variance, which is expected. If the expert has strong knowledge of the underlying process, the initial model bears little importance as the strong prior dominates the posterior. With lower values of k, there is more uncertainty in which local optimum, of which there can be multiple, is found as the navigation mainly depends on the candidate models' fit to the data. Navigation in the space of causal models may be initiated from any graph, for example we could always use an empty DAG as the initial model. The results, however, show that the choice of initial model can affect which model is obtained. As the final model can be different depending on the initial model, beginning from an empty graph may not always produce optimal results.

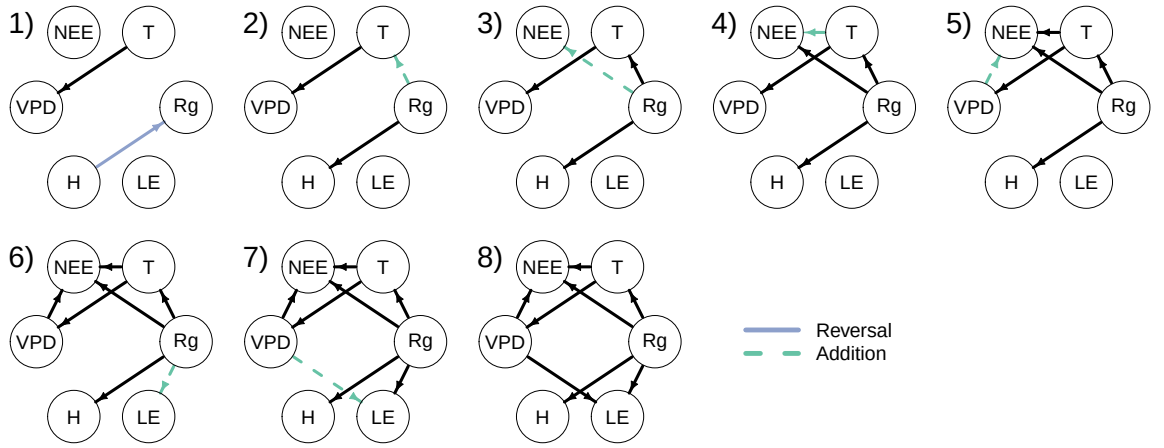


Figure 7.3: An example of an expert user navigation with data for April 2014.

7.2 Real-world Use Cases

The real-world data set is a set of measurements collected at the SMEAR II (System for Measuring Forest Ecosystem-Atmosphere Relationships II) station at Hyytiälä, Finland from 2013 to 2015 [59]. The data are part of the FLUXNET2015 data set [65] and the measurements are averaged at half hour intervals. Variables included in the analysis are shortwave downward radiation (Rg), air temperature (T), vapour pressure deficit (VPD), sensible heat flux (H), latent heat flux (LE), and net ecosystem exchange (NEE). A detailed characterisation of the variables is outside our scope but further information is available in a collection of research on forest ecology [32, 103]. We keep observations where the potential shortwave downward radiation is at least 80% of the daily maximum to mitigate effects of diurnal variation on the data distribution [51]. Additionally, measurements with gap filled values for NEE, H, or LE are filtered out to avoid introducing artificial causal relations. After filtering, the data set contains 817 data points for April 2013–2015, 854 data points for May 2013–2015, and 215 data points for August 2015. We perform the analysis on three combinations of months—April, April & May, and April & August—and varying number of years for April.

In each case, the highest scoring model output by the default set of four algorithms, PC-Stable with significance levels 0.01 and 0.1, GES, and LiNGAM, is used as an initial point for the navigation. We use k-fold blocked cross-validation to obtain training and validation scores for the graphs by splitting the data into k blocks of consecutive data points. Each block is then used as a validation set in one iteration while the model is trained on the remaining blocks. The number of blocks varies between use cases but, depending on the subset of the available data used, a block covers a period of approximately half a month, a month, or two months. Training score is the $\overline{R}_a^2$ over

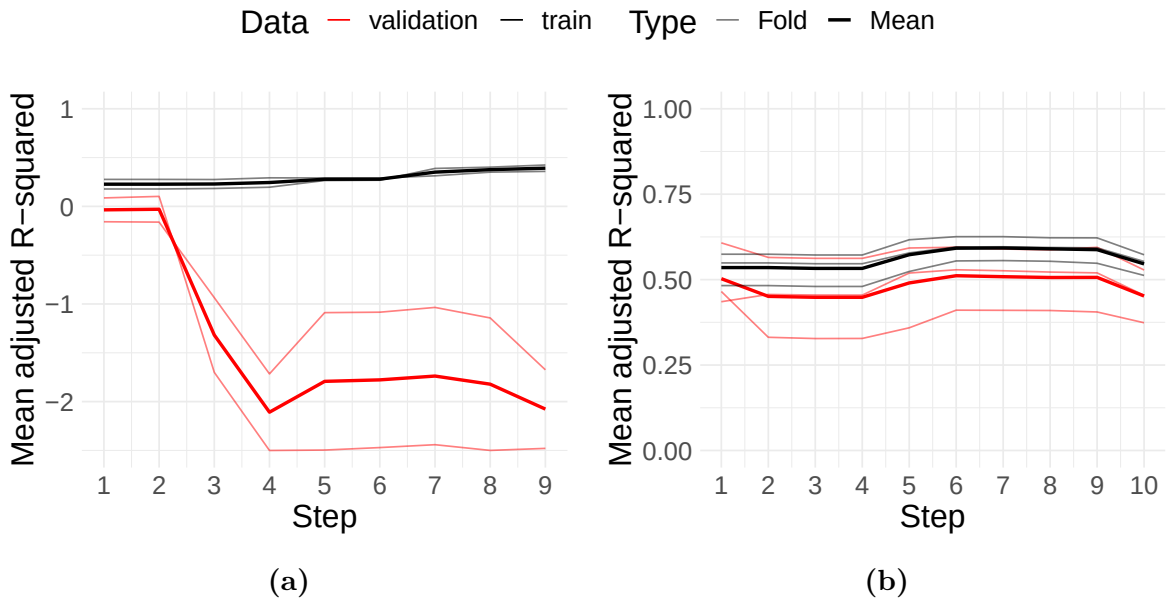


Figure 7.4: Use case 1. Showing the user the validation and training scores allows them to (a) detect and (b) mitigate overfitting problems. Validation and training $\overline{R}_a^2$ with data from (a) April 2014; (b) April and May 2013–2015. Note the different scaling of the y-axes.

the training data averaged over the k folds. Similarly, we compute the validation score as the mean $\overline{R}_a^2$ over the CV folds using the trained model to predict values for the validation set.

Figure 7.3 displays an example of how an expert user may edit graphs to navigate in the space of causal models. The trajectory of model scores for the navigation are shown in Figure 7.4a. The final model 8 in the figure has been provided by two domain area experts and the initial model 1 was obtained from the default set of algorithms. In use cases 1 and 2, model 8 is used as the target model in which the navigation ends beginning from an initial model. However, the initial models in the use cases are not equal.

7.2.1 Use Case 1: Detection of Overfitting

Overfitting is a common problem in modelling although, to the best of our knowledge, it has not been addressed in previous work in the context of interactive CSD. In this use case, we demonstrate how the user can detect overfitting by inspecting the training and validation scores as well as differences between them with 2-fold blocked CV on data measured in April 2014. Already the initial model, the best model output by the default algorithms as measured by $\overline{R}_a^2$, has a negative validation score which is shown in Figure 7.4a. As discussed in Subsection 6.2, a negative validation score indicates the mean of the validation data produces better predictions than the trained model.

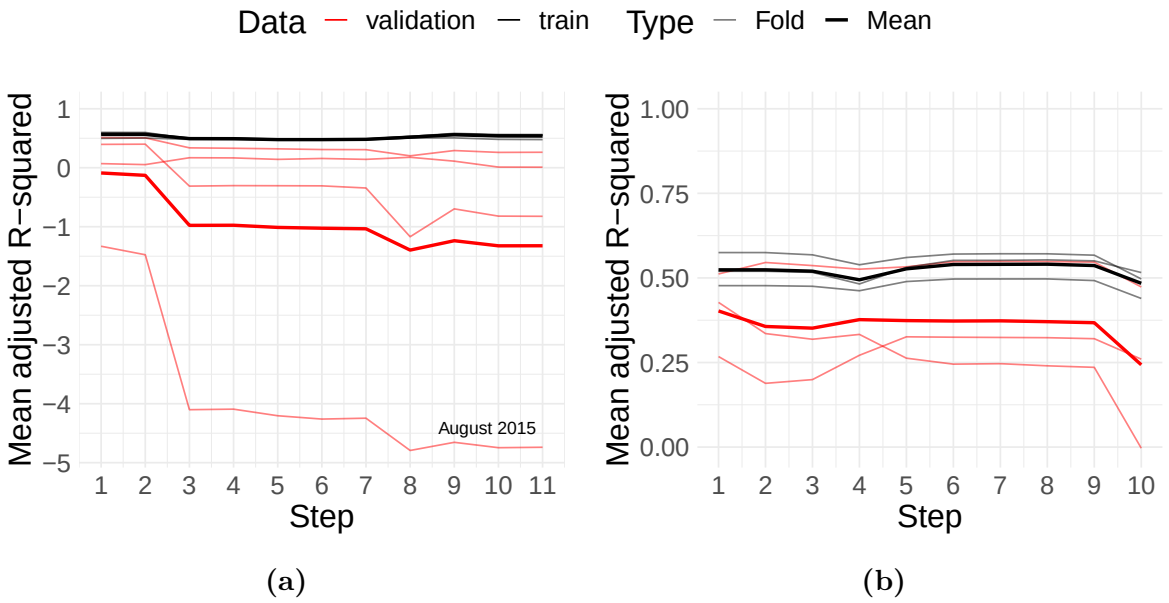


Figure 7.5: Use case 2. Results from analysis with data from (a) April 2013–2015 and August 2015, (b) April 2013–2015. Concept drift can be detected by cross-validation and resolved by leaving out the data with a different distribution. Note the different scaling of the y-axes.

After the model is edited, the training and validation scores diverge radically. Negative validation scores throughout the navigation can indicate overfitting or concept drift, and further investigation is required to determine the cause of the problem. Because the data contain samples from one month only, a likely issue is overfitting: the model specialises on the training data leading to inability to predict the validation data well. Once we add data to cover both April and May in 2013 through 2015, the validation score stays clearly positive for all three cross-validation folds and the training and validation score averages follow the same pattern throughout the analysis, as shown in Figure 7.4b.

Cross-validation is a well-known technique for controlling overfit, although we are not aware of its application in interactive CSD. This use case shows how CV can be used in interaction with the user. Without checking models for overfitting, we risk obtaining a model that does not generalise or does not reflect the true phenomena in the data. Showing the user validation and training scores enables them to decide whether the model has overfit the data.

7.2.2 Use Case 2: Detection of Concept Drift

Another problem that can arise in causal modelling with real-world data is concept drift. To demonstrate how the user can detect concept drift with blocked cross-validation, we analyse a data set containing samples from April in 2013–2015 and

(a) **(b)**

Figure 7.6: Result of navigation without any prior knowledge of the model when beginning from (a) the algorithm output with highest score and (b) an empty graph.

from August 2015. For one of the four folds, the score is negative already for the initial model and the mean validation score falls below zero after two edits. Furthermore, the validation and training score means diverge after the second edit. We notice that although the validation score is negative for more than one of the folds, the score for the fold containing the August data is clearly inferior to the rest, which suggests potential concept drift. Leaving out the problematic August data and repeating the analysis improves the scores significantly, leading to similar, non-negative trajectories for the training and validation scores, as shown in Figure 7.5b.

This use case illustrates how the user can detect concept drift that can occur in real-world systems. Undetected concept drift can result in a model that fits none of the subsets of data well. Problematic subsets of the data can be identified by the user by displaying to them information on the validation and training scores for each of the cross-validation folds consisting of contiguous data blocks.

7.2.3 Use Case 3: Effect of Initial Model

Depending on the choice of initial model, different models that fit the data approximately equally well can be found. When the expert has strong knowledge of causal relationships between all pairs of variables, the initial model has little impact on the final model as the strong prior dominates the posterior. However, when the user has little or no knowledge of the data-generating process, results are more sensitive to variations in the initial model than with high levels of knowledge due to the greedy approach to finding a local optimum of the posterior. To demonstrate this idea, we assume a flat user prior and begin navigation both from the highest scoring output obtained from the default set of CSD algorithms and from an empty graph. In each step, we greedily navigate to the neighbour with highest score and stop once the score cannot be improved by a single edit. The final models obtained with no knowledge and

different initial models are displayed in Figure 7.6. A slightly different local optimum of the approximate posterior is reached depending on the initial model. Even though the two models are rather similar with a structural Hamming distance (SHD) of two, the example highlights the possibility of finding a different model that fits the data equally well or better when changing the initial model. Although a good result from our approach does not necessarily resemble a "true" model as the aim is to find a local optimum of the user's posterior, we note here that the SHD between the experts' model 8 in Figure 7.3 and the final models in Figure 7.6 are seven and five, respectively.

8. Discussion

In this thesis, we have illustrated through experiments with simulated rational user and use cases with real-world data how an expert user's prior knowledge can be incorporated into causal structure discovery (CSD) through interactions. Our focus has been on discovering causal model structures from observational data but we propose a Bayesian formulation of the problem. The models we discuss are Bayesian networks that are assigned with a causal interpretation which is not necessary for the validity of the proposed procedure. Hence, the conclusions we present can be applied more generally to Bayesian model building.

Our results show how adding even low levels of background knowledge the outputs from CSD algorithms can be improved in terms of fitting both the data and the expert's prior knowledge. Because our formulation of the process relies on a greedy optimisation of the user's approximate posterior, the choice of initial model for the navigation can affect the final result if the approximate posterior contains multiple local optima. With cross-validation, problems of overfitting and concept drift can be detected which enables the expert user to investigate the causes of such issues.

We provide a Bayesian formulation of the problem underlying CSD and our procedure for performing the analysis partly fits the Bayesian workflow [28]. The main differences are that we do not have access to the user's prior distribution and we are interested only in the model structure and not in the remaining model parameters. From the Bayesian workflow, our approach contains the modules of choosing an initial model, fitting and evaluating a model, modifying the model, and comparing a number of models.

Comparisons among models are performed both when the expert chooses an initial model from the algorithm outputs as well as during the analysis when they make decisions regarding which model to navigate to. Although we leave out causal effect size estimation from the workflow, model fitting is performed to provide the expert with estimates on goodness-of-fit of the model structures. For the estimates, we use cross-validation to evaluate the model to efficiently use the data and to enable detection of problems in the modelling process. During the navigation, the expert iteratively modifies the current model to move to the best one of the neighbouring models with

respect to their prior knowledge and goodness-of-fit.

Apart from the initial model selection, models are compared only with their neighbours. The expert can perform the navigation multiple times beginning from different initial models and then compare the final results. A possible extension to our current approach would be to allow side-by-side navigation and comparisons among models with different initial points instead of comparing only the final models. With multiple concurrently edited models, the expert could inspect the number of models where each edge is present or absent and use that information to construct an aggregate causal model. Alternatively, seeing multiple models simultaneously can help the expert distinguish between them better than by inspecting each model alone.

Our solution relies on a number of assumptions which may not be necessary with a different approach to apply the suggested procedure to interactive CSD. We assume linear relationships and Gaussianity of noise in the data: the score we use to evaluate model fit and provide approximation of the model's log-likelihood is based on linear regression and is proportional to log-likelihood only under linearity and Gaussianity. However, real-world data sets cannot be expected to follow such constraints faithfully and a different set of assumptions could produce different results. The carbon dioxide flux data set we analysed in the use cases is an example of a data set where Gaussianity and linearity are not fully met and the domain experts' input is essential to counteract the negative effects of invalid assumptions. It would be interesting to study how exactly the results are affected by the choice of assumptions regarding the functional family and noise distribution of the data. One option for a more generally applicable framework would be to use non-parametric approaches to likelihood estimation, such as Gaussian processes [114] or kernel conditional density estimation [41].

Regarding the user, we assume that an expert user can be modelled as a rational Bayesian agent, which is a strong albeit commonly used assumption [e.g., 14, 104]. Even though people have been found to act in an approximately Bayesian manner, they tend to overestimate the posterior of events with low true probabilities and underestimate the posterior of events with high true probabilities [109]. Prior research has shown that to express their beliefs, people often use biased heuristics which do not conform to the Bayes' rule [102]. On the other hand, even though people are not, in general, fully Bayesian, the Bayes' rule has been found to be the most likely rule they follow in decision making [20]. We assumed in our formulation only that the experts act as Bayesian rational agents. In future research, more complex user models could be employed to model the user's biases in addition to their knowledge. For example, resource rationality refers to modelling human cognition as the rational use of limited cognitive resources [56]. It has been proposed as an alternative to biased heuristics for explaining the cognitive behaviour of people which can be suboptimal in terms

of rationality. Such approaches that take into consideration cognitive limitations of people could help provide estimates of uncertainty in the found causal models.

In the field of interactive visualisation, a study has found that people perform approximate Bayesian inference when averaged over a group of individuals although their personal estimates are biased [47]. With large sample sizes, the aggregated estimates deviate from the fully Bayesian estimates although aggregation is found useful with small data sets [47]. These findings provide motivation for future research into generalising our approach to allow multiple experts to aggregate and incorporate their prior beliefs into the causal structure discovery through interactions.

The data set from our use cases consists of a number of measurements over time although time was not included as a variable in the analysis. Taking into account the temporal dimension in causal model building would enable the discovery of lagged causal relations. For example, a headache is not cured immediately on the ingestion of pain medication but the effect can be observed after a short period of time. Allowing lagged effects enables the modelling of feedback loops without concurrent cycles in the models. The interactions we have proposed would be valid likewise with lagged connections but the visualisation of the causal models would require some changes. Often temporal causal models are visualised as gridded graphs where each row corresponds to a model variable and each column to a time step with the current time t as the rightmost column and number of columns equal to maximum number of lags allowed [79]. A similar visualisation could be included in interactive CSD although the adjacency matrix through which interactions are performed in our approach would require some changes.

Our proposed procedure focuses solely on identifying a causal structure for the process of interest. However, with the current implementation it would be reasonably simple to estimate the effect sizes for the found causal relations. Under the assumptions of linearity and Gaussianity, the effect sizes can be estimated as the coefficients from linearly regressing variables on their parents [58]. Different methods would need to be deployed together with clear definitions of effect sizes if a different set of assumptions were used. Another consideration for effect size estimation stems from the incorporation of expert knowledge. Eliciting numeric estimations from experts tends to result in more bias than the simpler task of eliciting beliefs regarding statistical independence relations [25]. Furthermore, our proposed interactions are insufficient for eliciting effect size estimations from the user. More fine-grained types of interactions would be needed to allow the expert to input their beliefs of the effect sizes.

We use a number of CSD algorithms to provide the expert user with a selection of initial models for the navigation. In our formulation, the initial models are a sample from the computer's posterior distribution but other methods for finding initial models

exist and could be used here. Considering the Bayesian aspects of our proposal, employing Bayesian methods for the first step of the procedure would seem natural. For example, Markov chain Monte Carlo for structure learning of Bayesian networks has received attention recently [e.g., 52, 53, 108]. Our approach is easily extendable to new methods for providing the initial models and one topic for future research would be to study and compare how the methods used for the initial CSD affect the final results.

As discussed in Subsection 4.3, multiple cross-validation methods exist for data with temporal dependencies. The choice of method depends on properties of the available data and different methods can produce different results. Here, we have tested our approach using only one cross-validation method for dependent data and a separate one for dependent data. We did not perform experiments to determine whether our choices are optimal and further research is needed to study the effect the choice of cross-validation method has on the results.

In addition to building models to understand the processes of interest, our procedure can be used to iteratively perform feature selection. By building causal models for a process, we can find which variables act as drivers for a chosen target variable. With the set of drivers and the target as model variables, we can perform the interactive analysis again to discover the causal structure between them. Running causal analysis multiple times on the same data but with different variables could help the expert understand the underlying processes better.

Finally, our experiments and use cases serve to exemplify both how the approach could be used in practice and that the results are sensible in theory. The setting in the simulated user experiments does not, however, represent fully how the method would be used by field experts. User studies are needed to determine the usefulness of the approach in practical settings. The evaluation of the results poses a difficult problem in real-world circumstances as we cannot compare them with the user's posterior distribution.

9. Conclusions

Previous work in adding interactivity to causal structure discovery (CSD) has focused on building systems for domain experts rather than exploring the theoretical framework for the process. When building an end product, the system has to take into account the intended application area which usually encompasses making a set of assumptions that may not be valid for other problems. Our goal was to develop a high-level, modular workflow that can be applied to a variety of problems by changing the exact implementation to fit the circumstances. In this thesis, we have proposed such a procedure for interactive CSD and one manner of implementing the solution in practice. We have focused on applying interactive CSD to problems in Earth system sciences but the workflow remains valid for data sets pertaining to other fields.

The procedure consists of three parts—discovery of the set of initial models, navigation in the space of causal models, and validation for model selection and evaluation—whose exact formulation is not fixed. Through simulated user experiments and use cases with real-world data we have shown how incorporating a domain expert's prior knowledge into causal analysis and providing the expert with multiple initial models to choose from can improve the results. By improvement we refer to finding models that fit both the data and the domain expert's prior knowledge better than without combining automated methods with expert interactions. We demonstrated in the use cases how cross-validation can help detect overfitting and concept drift which are commonly occurring problems in real-world settings where the sample size of available data is limited.

Automated methods alone are often unable to establish the orientation of a causal relation even when its existence can be determined. Some algorithms, such as those based on structural equation models, are able to distinguish between the two possible directions between causally linked variables by assuming independent and non-Gaussian noise. Apart from LiNGAM, the algorithms we used to discover the set of initial models assume Gaussian noise distributions. When the data adhere to the Gaussian assumption, most orientations of edges during the navigation represent the expert's prior knowledge rather than information that can be ascertained from the data. Real-world data, however, rarely meets such assumptions fully.

As discussed in the previous Section 8, many questions and possibilities for extensions in interactive CSD still remain open. Our problem formulation and its practical implementation form just one alternative for manifesting the workflow. A different set of assumptions would lead to other solutions. User studies would help determine the steps that are needed to ensure the practical applicability of our approach. In order to build interactive systems for use in research, different methods for implementing the workflow need to be explored and validated in different settings and application areas.

Evaluating the results of interactive model building is non-trivial. The found models cannot be compared with some ground truth as the goal of the process is to find models that represent not only the best fit to data but also the background knowledge of the expert. In this work, we were able to perform model comparisons in the simulated user studies because of the model we assumed for the user. We did not consider incorrect knowledge and, thus, comparisons with the true model of the synthetic data were possible. In real-world application, knowledge that is contrary to the true model cannot be ruled out and, furthermore, a true model is often not available.

Even though further research is needed to explore the optimal implementations for the workflow, our experiments serve to demonstrate the potential of the approach in practical applications. The results show it is useful to incorporate a domain expert's prior knowledge into causal analysis by allowing interactions as part of causal structure discovery. Providing the expert with multiple initial models to choose from is found similarly useful and cross-validation helps the expert detect overfitting and concept drift when analysing finite data sets while allowing efficient use of the data to build models. Despite the issues and disclaimers discussed in this and the previous chapter, our approach seems reasonable and shows potential for systems of interactive causal structure discovery.

Bibliography

[1] F. J. Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, May 1960.

[2] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, January 2010.

[3] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, Cambridge, England, 2012.

[4] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, December 2004.

[5] C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, May 2012.

[6] C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, April 2018.

[7] C. R. Blyth. On Simpson's paradox and the sure-thing principle. *Journal of the American Statistical Association*, 67(338):364–366, June 1972.

[8] P. Burman, E. Chow, and D. Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, June 1994.

[9] J. Y. Campbell and S. B. Thompson. Predicting excess stock returns out of sample: Can anything beat the historical average? *Review of Financial Studies*, 21(4):1509–1531, July 2008.

[10] D. M. Chickering. Learning equivalence classes of Bayesian network structures. In E. Horvitz and F. Jensen, editors, *International Conference on Uncertainty in Artificial Intelligence*, UAI'96, pages 150–157. Morgan Kaufmann Publishers Inc., August 1996.

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

[12] D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3921–3962, January 2014.

[13] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, April 1994.

[14] P. Daee, T. Peltola, A. Vehtari, and S. Kaski. User modelling for avoiding overfitting in interactive knowledge elicitation for prediction. In S. Berkovsky, Y. Hijikata, J. Rekimoto, M. M. Burnett, M. Billinghurst, and A. Quigley, editors, *International Conference on Intelligent User Interfaces*, IUI'18, pages 305–310. Association for Computing Machinery, March 2018.

[15] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, Cambridge, England, 2009.

[16] M. de Jongh and M. J. Druzdzel. *A comparison of structural distance measures for causal Bayesian network models*, pages 443–456. Academic Publishing House EXIT, Warsaw, Poland, 2009.

[17] Y. Deng and I. Ebert-Uphoff. Weakening of atmospheric information flow in a warming climate in the community climate system model. *Geophysical Research Letters*, 41(1):193–200, January 2014.

[18] I. Ebert-Uphoff and Y. Deng. A new type of climate network based on probabilistic graphical models: Results of boreal winter versus summer. *Geophysical Research Letters*, 39(19), October 2012.

[19] I. Ebert-Uphoff and Y. Deng. Identifying physical interactions from climate data: Challenges and opportunities. *Computing in Science & Engineering*, 17(6):27–34, November 2015.

[20] M. A. El-Gamal and D. M. Grether. Are people Bayesian? Uncovering behavioral strategies. *Journal of the American Statistical Association*, 90(432):1137–1145, December 1995.

[21] M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro. Incorporating expert knowledge when learning Bayesian network structure: A medical case study. *Artificial Intelligence in Medicine*, 53(3):181–204, November 2011.

[22] A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(3):350–363, July 1972.

[23] J. a. Gama, I. Žliobaité, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, March 2014.

[24] J. A. Gámez, J. L. Mateo, and J. M. Puerta. A fast hill-climbing algorithm for Bayesian networks structure learning. In K. Mellouli, editor, *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU'07, pages 585–597. Springer-Verlag, October 2007.

[25] P. H. Garthwaite, J. B. Kadane, and A. O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–701, June 2005.

[26] X. Ge, V. K. Raghu, P. K. Chrysanthis, and P. V. Benos. CausalMGM: An interactive web-based causal discovery tool. *Nucleic Acids Research*, 48(W1):W597–W602, July 2020.

[27] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC texts in statistical science series. CRC Press, Boca Raton, FL, USA, 3rd edition, 2014.

[28] A. Gelman, A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, L. Kennedy, J. Gabry, P.-C. Bürkner, and M. Modrák. Bayesian workflow, November 2020. arXiv:2011.01808 [stat.ME].

[29] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, January 1992.

[30] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, August 1969.

[31] J. D. Hamilton. *Time series analysis*, chapter 11.2, pages 302–308. Princeton University Press, Princeton, NJ, USA, 1994.

[32] P. Hari, K. Heliövaara, and L. Kulmala. Glossary. In P. Hari, K. Heliövaara, and L. Kulmala, editors, *Physical and Physiological Forest Ecology*, pages 521–530. Springer, London, UK, December 2012.

[33] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, chapter 7, pages 219–257. Springer series in statistics. Springer, New York, NY, USA, 2nd edition, 2009.

[34] A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464, August 2012.

[35] D. M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, January 2004.

[36] D. Heckerman and D. Geiger. Likelihoods and parameter priors for Bayesian networks. Technical Report MSR-TR-95-54, Microsoft Research, Redmond, WA, USA, November 1995.

[37] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, September 1995.

[38] P. W. Holland. Causal inference, path analysis, and recursive structural equations models. *Sociological Methodology*, 18:449–484, 1988.

[39] M. N. Hoque and K. Mueller. Outcome-Explorer: A causality guided interactive visual interface for interpretable algorithmic decision making, January 2021. arXiv:2101.00633v2 [cs.HC].

[40] P. O. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *International Conference on Neural Information Processing Systems*, NIPS'08, pages 689–696. Curran Associates Inc., December 2008.

[41] R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4):315–336, December 1996.

[42] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning: with applications in R.* Springer texts in statistics. Springer, New York, NY, USA, 2013.

[43] Z. Jin, S. Guo, N. Chen, D. Weiskopf, D. Gotz, and N. Cao. Visual causality analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 27:1343–1352, February 2021.

[44] M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, May 2012.

[45] R. K. Kaufmann and D. I. Stern. Evidence for human influence on climate from hemispheric temperature relations. *Nature*, 388(6637):39–44, July 1997.

[46] R. J. Kennett, K. B. Korb, and A. E. Nicholson. Seabreeze prediction using Bayesian networks. In D. W.-L. Cheung, G. J. Williams, and Q. Li, editors, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD'01, pages 148–153. Springer-Verlag, April 2001.

[47] Y.-S. Kim, L. A. Walls, P. Krafft, and J. Hullman. A Bayesian cognition approach to improve data visualization. In S. Brewster and G. Fitzpatrick, editors, *Conference on Human Factors in Computing Systems*, CHI'19, pages 1–14. Association for Computing Machinery, May 2019.

[48] E. Kodra, S. Chatterjee, and A. R. Ganguly. Exploring Granger causality between global average observed time series of carbon dioxide and temperature. *Theoretical and Applied Climatology*, 104(3-4):325–335, July 2011.

[49] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In U. M. Fayyad and R. Uthurusamy, editors, *International Conference on Knowledge Discovery and Data Mining*, KDD'95, pages 192–197. AAAI Press, August 1995.

[50] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. Adaptive computation and machine learning. MIT Press, Cambridge, MA, USA, 2009.

[51] C. Krich, J. Runge, D. G. Miralles, M. Migliavacca, O. Perez-Priego, T. El-Madany, A. Carrara, and M. D. Mahecha. Estimating causal networks in biosphere-atmosphere interaction with the PCMCI approach. *Biogeosciences*, 17(4):1033–1061, February 2020.

[52] J. Kuipers and G. Moffa. Partition MCMC for inference on acyclic digraphs. *Journal of the American Statistical Association*, 112(517):282–299, January 2017.

[53] J. Kuipers, P. Suter, and G. Moffa. Efficient sampling and structure learning of Bayesian networks, January 2020. arXiv:1803.07859v3 [stat.ML].

[54] J. A. Landsheer. The specification of causal models with Tetrad IV: A review. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(4):703–711, October 2010.

[55] D. Lewis. Causation. *The Journal of Philosophy*, 70(17):556–567, October 1973.

[56] F. Lieder and T. L. Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1, February 2019.

[57] J. Liu and D. Niyogi. Identification of linkages between urban heat island magnitude and urban rainfall modification by use of causal discovery algorithms. *Urban Climate*, 33, September 2020.

[58] M. H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, December 2009.

[59] I. Mammarella. Drought 2018 fluxdata preview selection, Hyytiälä, 1995-12-31–2018-12-31, March 2020.

[60] C. Meek. Causal inference and causal explanation with background knowledge. In P. Besnard and S. Hanks, editors, *Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 403–410. Morgan Kaufmann Publishers Inc., August 1995.

[61] L. Melkas, R. Savvides, S. Chandramouli, J. Mäkelä, T. Nieminen, I. Mammarella, and K. Puolamäki. Interactive causal structure discovery in Earth system sciences. Submitted in May 2021.

[62] H. E. Niles. Correlation, causation and Wright's theory of "path coefficients". *Genetics*, 7(3):258–273, May 1922.

[63] P. Nowack, J. Runge, V. Eyring, and J. D. Haigh. Causal networks for climate model evaluation and constrained projections. *Nature Communications*, 11(1), December 2020.

[64] R. T. O'Donnell, A. E. Nicholson, B. Han, K. B. Korb, M. J. Alam, and L. R. Hope. Causal discovery with prior information. In A. Sattar and B.-h. Kang, editors, *Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, AI'06, pages 1162–1167. Springer-Verlag, December 2006.

[65] G. Pastorello et al. The FLUXNET2015 dataset and the ONEFlux processing pipeline for eddy covariance data. *Scientific Data*, 7(1):225, July 2020.

[66] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. Technical Report CSD-850021, R-43, Cognitive Systems Laboratory, Computer Science Department, UCLA, June 1985.

[67] J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–710, December 1995.

[68] J. Pearl. *Causality*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

[69] J. Pearl and D. MacKenzie. *The book of why: The new science of cause and effect*. Basic Books, New York, NY, USA, 1st edition, 2018.

[70] K. Pearson. *The Grammar of Science*, chapter 4. Adam and Charles Black, London, England, 2nd edition, 1900.

[71] A. Pérez-Suay and G. Camps-Valls. Causal inference in geoscience and remote sensing from observational data. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1502–1513, March 2019.

[72] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2020.

[73] J. Racine. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99(1):39–61, November 2000.

[74] J. Ramsey, S. Hanson, C. Hanson, Y. Halchenko, R. Poldrack, and C. Glymour. Six problems for causal inference from fMRI. *NeuroImage*, 49(2):1545–1558, January 2010.

[75] J. D. Ramsey. Scaling up greedy causal search for continuous variables, November 2015. arXiv:1507.07749v2 [cs.AI].

[76] J. D. Ramsey, P. Spirtes, and J. Zhang. Adjacency-faithfulness and conservative causal inference. In R. Dechter and T. Richardson, editors, *Conference on Uncertainty in Artificial Intelligence*, UAI'06, pages 401–408. AUAI Press, July 2006.

[77] R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in the Theory of Graphs*, pages 239–273. Academic Press, New York, NY, USA, January 1973.

[78] J. Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In J. Peters and D. Sontag, editors, *Conference on Uncertainty in Artificial Intelligence*, volume 124 of *UAI'20*, pages 1388–1397. PMLR, August 2020.

[79] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz Marí, E. H. van Nes, J. Peters, R. Quax, M. Reichstein, M. Scheffer, B. Schölkopf, P. Spirtes, G. Sugihara, J. Sun, K. Zhang, and J. Zscheischler. Inferring causation from time series in Earth system sciences. *Nature Communications*, 10(1), December 2019.

[80] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11), November 2019.

[81] S. M. Samarasinghe, E. A. Barnes, and I. Ebert-Uphoff. Causal discovery in the presence of confounding latent variables for climate science. In C. Chen, D. Cooley, J. Runge, and E. Szekely, editors, *International Workshop on Climate Informatics: CI 2018*, volume NCAR/TN-550+PROC of *NCAR Techical Notes*, pages 53–56. National Center for Atmospheric Research, November 2018.

[82] S. M. Samarasinghe, M. C. McGraw, E. A. Barnes, and I. Ebert-Uphoff. A study of links between the Arctic and the midlatitude jet stream using Granger and Pearl causality. *Environmetrics*, 30(4), June 2019.

[83] C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10(2):153–178, February 1993.

[84] R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The TETRAD project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, 33(1):65–117, January 1998.

[85] J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In T. Kehler, editor, *AAAI National Conference on Artificial Intelligence*, AAAI'86, pages 502–507. AAAI Press, August 1986.

[86] J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):486–494, June 1993.

[87] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, December 2006.

[88] H. A. Simon. On the definition of the causal relation. *The Journal of Philosophy*, 49(16):517–528, July 1952.

[89] H. A. Simon. Spurious correlation: A causal interpretation. *Journal of the American Statistical Association*, 49(267):467–479, September 1954.

[90] E. H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, July 1951.

[91] D. A. Smirnov and I. I. Mokhov. From Granger causality to long-term causality: Application to climatic data. *Physical Review E*, 80, July 2009.

[92] M. Smithson. Correct confidence intervals for various regression effect sizes and parameters: The importance of noncentral distributions in computing intervals. *Educational and Psychological Measurement*, 61(4):605–632, August 2001.

[93] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, April 1991.

[94] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*. Adaptive computation and machine learning. MIT Press, Cambridge, MA, USA, 2nd edition, 2000.

[95] H. O. Stolberg, G. Norman, and I. Trop. Randomized controlled trials. *American Journal of Roentgenology*, 183(6):1539–1544, December 2004.

[96] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, January 1974.

[97] N. R. Swanson and H. White. A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. *Journal of Business & Economic Statistics*, 13(3):265–275, July 1995.

[98] L. J. Tashman. Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4):437–450, October 2000.

[99] R. E. Tillman, A. Gretton, and P. Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *International Conference on Neural Information Processing Systems*, NIPS'09, pages 1847–1855. Curran Associates Inc., December 2009.

[100] U. Triacca. Is Granger causality analysis appropriate to investigate the relationship between atmospheric concentration of carbon dioxide and global surface air temperature? *Theoretical and Applied Climatology*, 81(3-4):133–135, July 2005.

[101] 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, October 2006.

[102] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, September 1974.

[103] Üllar Rannik, S. Launiainen, J. Pumpanen, L. Kulmala, P. Kolari, T. Vesala, J. F. J. Korhonen, and P. Hari. Environmental factors. In P. Hari, K. Heliövaara, and L. Kulmala, editors, *Physical and Physiological Forest Ecology*, chapter 3, pages 27–46. Springer, London, UK, December 2012.

[104] E. Van den Steen. Overconfidence by Bayesian-rational agents. *Management Science*, 57(5):884–896, May 2011.

[105] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1), February 2006.

[106] T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. In R. Shachter, T. Levitt, L. Kanal, and J. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 9 of *Machine Intelligence and Pattern Recognition Series*, pages 69–76. North-Holland, Amsterdam, the Netherlands, June 1990.

[107] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In P. Bonissone, M. Henrion, L. Kanal, and J. Lemmer, editors, *Annual Conference on Uncertainty in Artificial Intelligence*, UAI'90, pages 255–270. Elsevier Science Inc., July 1990.

[108] J. Viinikka, A. Hyttinen, J. Pensar, and M. Koivisto. Towards scalable Bayesian learning of causal DAGs. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33 of *NeurIPS'20*, pages 6584–6594. Curran Associates, Inc., December 2020.

[109] W. K. Viscusi. Are individuals Bayesian decision makers? *The American Economic Review*, 75(2):381–385, May 1985.

[110] C. S. Wallace, K. B. Korb, and H. Dai. Causal discovery via MML. In L. Saitta, editor, *International Conference on Machine Learning*, ICML'96, pages 516–524. Morgan Kaufmann Publishers Inc., July 1996.

[111] J. Wang and K. Mueller. The visual causality analyst: An interactive interface for causal reasoning. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):230–239, January 2016.

[112] J. Wang and K. Mueller. Visual causality analysis made practical. In B. Fisher, S. Liu, and T. Schreck, editors, *IEEE Conference on Visual Analytics Science and Technology*, VAST'17, pages 151–161. IEEE, October 2017.

[113] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, April 1996.

[114] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *International Conference on Neural Information Processing Systems*, NIPS'95, pages 514–520. MIT Press, November 1995.

[115] S. Wright. Correlation and causation. *Journal of Agricultural Research*, 20(7):557–585, January 1921.

[116] K. Zhang and A. Hyvärinen. Distinguishing causes from effects using nonlinear acyclic causal models. In I. Guyon, D. Janzing, and B. Schölkopf, editors, *International Conference on Causality: Objectives and Assessment*, volume 6 of *COA'08*, pages 157–164. JMLR.org, December 2008.