TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Department of Mathematics and Computer Science
Statistics Group

Structure Learning in High-Dimensional Time Series Data

Master Thesis

Martin de Quincey

Supervisors:
dr. Rui Castro
dr. Alex Mey

Assessment Committee Members:
dr. Rui Castro
dr. Alex Mey
dr. Jacques Resing

version 0.4

Eindhoven, June 2022

Contents

Chapter 3

Previous Work

In Chapter 1, we have introduced the notion of *structure learning* on a conceptual level. Several applications and motivating examples were given to underline the usefulness of learning a structure of the data matrix $\mathbf{X}$. In Chapter 2, the objective of this thesis has been made more explicit with corresponding mathematical notation. We are interested in learning the joint probability of the data matrix $\mathbf{X}$, casting the problem to a *density estimation* task. To learn a suitable joint probability, several assumptions have been made on the factorization of the joint probability, namely that each variable X_i only depends on its parent set $\mathrm{Pa}\left(X_i\right)$. Now, there are several different methods to learn such a suitable factorization, which can be decomposed into three main categories.

The first main category was also the first method used for structure learning, first proposed by Verma and Pearl in [63], where edge constraints were derived using conditional independence test on multiple subsets of variables, together with their developed framework of causal inference. These edge constraints together formed a skeleton of undirected edges, from which directed relationships could be deduced. These methods can be categorized as *constraint-based* approaches, which are discussed in greater detail in Section 3.1.

A second type of approach to learn a directed acyclic structure is to exploit asymmetries in the noise. Without any additional assumptions on the data, determining the direction of instantaneous relations between variables is impossible [86]. This means that we can detect that there is an arc in either direction, so $X_i - X_j$, yet it is impossible to determine whether $X_i \rightarrow X_j$ or $X_i \leftarrow X_j$. Interestingly, if we make some assumptions about the distribution of the noise variables in $\mathbf{X}$, we can determine the directionality of the arc. As these methods rely on assumptions about the noise variables, this type of approach is called *noise-based* approaches. A deeper explanation of the principles behind these noise structure-based approaches, as well as some methods that employ this type of approach, will be discussed in Section 3.2.

The third type of category is named *score-based* approaches, which has seen the most developments in the past years. Score-based approaches assess the validity of a given structure $\mathcal{G}$ by assigning, as the name suggests, a score to each structure based on how well $\mathcal{G}$ fits the data $\mathbf{X}$. This score is determined by a scoring function $S(\mathbf{X}, \mathcal{G})$. Consequently, the optimal structure is the structure $\mathcal{G}^*$ that maximizes the scoring function $S(\mathbf{X}, \mathcal{G})$, while remaining acyclic,

$$\mathcal{G}^* = \underset{\text{structures } \mathcal{G}}{\arg\max}\ S(\mathbf{X}, \mathcal{G}) \text{ such that } \mathcal{G} \in \texttt{DAGs}. \tag{3.1}$$

Unfortunately, maximizing such a score function over the set of directed acyclic graphs is NP-hard [16]. Therefore, we cannot expect to find G^* in polynomial time. Nevertheless, several interesting approaches have been developed that efficiently search for a suboptimal yet satisfactory structure. Furthermore, interesting techniques have been developed to find an optimal structure $\mathcal{G}^*$ as efficient as possible, albeit still exponential in running time. Several of these score-based approaches will be covered in Section 3.3.

3.1 Constraint-Based Approaches

Constraint-based approaches were among the first types of approach to identify causal relationships in the causal framework introduced by Pearl [60]. Constraint-based approaches consists of, as the name suggest, identifying *edge constraints* using conditional independence tests.

Identifying the skeleton. First, the skeleton of the structure is identified using conditional independence tests. As these conditional independence tests can only detect undirected relations, the skeleton contains only undirected edges. Let $V = \{X_1, \ldots, X_p\}$ be the set of all p variables. Then, there exists an undirected edge between two variables X_i and X_j if and only if they are conditionally independent for all subsets S not containing X_i and X_j,

$$\text{edge } (X_i, X_j) \text{ in skeleton of } \mathcal{G} \iff X_i \not\!\perp\!\!\!\perp X_j \mid S \qquad \forall\ S \subseteq V \setminus \{X_i, X_j\}. \tag{3.2}$$

A more intuitive approach the converse of Equation 3.2. If we can find a set S such that X_i and X_j are conditionally independent, then the undirected edge (X_i, X_j) is not included in the skeleton,

$$\exists\ S \subseteq V \setminus \{X_i, X_j\} : X_i \perp\!\!\!\perp X_j \mid S \Rightarrow \text{edge } (X_i, X_j) \text{ not in skeleton of } \mathcal{G}. \tag{3.3}$$

We can derive the skeleton of direct dependencies by starting with a fully connected undirected graph and removing edges using the rule in Equation 3.3. In real-life data, we can use any conditional independence test such as a Fisher-z's test [27] to determine whether the conditional dependence is significant. However, this requires such a large amount of conditional independence tests which causes issues as the accuracy of skeleton recovery decreases and could potentially yield conflicting results in the orientation of edges [49]. Furthermore, conditional independence tests require a large number of samples which may not always be available [71]. These are the main drawbacks of constraint-based approaches.

Identifying immoralities. Having identified the skeleton of our causal network, we still need to orient the edges. Constrained-based approaches make use of the fact the orientation of edges can be deduced from already derived conditional (in)dependencies and orientations. Firstly, we can orient edges edges by identifying *immoralities*. Suppose that we have discovered using independence tests that $X - Y - Z$, meaning that there is a direct dependency between X and Y and between Y and Z. Furthermore, assume that X and Z are independent conditioned on some subset that does not include Y, then we call the triple (X, Y, Z) an *immorality*, *v-structure*, or *collider* [55].

Now, as X and Z are independent given some set of variables not including Y, yet there is a direct dependency between both X and Y and Z and Y, we know that Y cannot cause X and Z. In fact, we know that X and Z both cause Y. Therefore, we can orient these immoralities as $X \rightarrow Y \leftarrow Z$.

Further orientation of edges. Once edges have been removed from the complete graph and all immoralities have been oriented using conditional independence tests, we can use a set of rules to learn the correct orientation of the edges. A simple set of just four rules that is proven to be both sound and complete is called the "Meek rules" [52]. The Meek rules orient edges such that we do not include any additional immoralities and we do not obtain a directed cycle in the structure.

Sometimes, deducing the correct orientation of the edge is not possible. Therefore, constraint-based approaches often output a graph that is maximally oriented, yet there are some edges that are undirected. Such a graph is called a *Completed Partially Directed Acyclic Graph* (CPDAG) and represents the *Markov Equivalence Class* (MEC) which contains all possible directed acyclic graphs that yield the same joint probability distribution, in the sense that they are statistically indistinguishable.

Pseudocode. The pseudocode for the constraint-based approach has been given in Algorithm 3.1. The two initial constraint-based approaches are SGS, proposed by Spirtes, Glymour, and Scheines [74], and Inductive Causation (IC), proposed by Verma and Pearl [61]. Although SGS technically outputs the set of possible directed acyclic graphs and IC would return the CPDAG, both outputs boil down to the exact same Markov Equivalence Class.

Improvements over these two algorithms have been proposed in the subsequent years. The PC-algorithm [75], named after Peter Spirtes and Clarke Glymour, two of the three authors of the SGS algorithm, returns the exact same output as the SGS or IC algorithm, but is significantly faster. The PC-algorithm differs in how it approaches step 2. Instead of trying all possible sets $S \subseteq V \setminus \{X_i, X_j\}$, it uses a more efficient way by first trying smaller sets of variables adjacent to X_i and X_j, such that the skeleton can be discovered using fewer independence tests.

Algorithm 3.1: Pseudocode of the constraint-based approach.

Input: A data matrix $\mathbf{X} = [X_1, X_2, \ldots, X_p]$.
Output: A maximally oriented CPDAG $\mathcal{G}$, corresponding to a set of statistically indistinguishable causal models of $\mathbf{X}$.

```
1: let $\mathcal{G}$ be a complete undirected graph
2: remove all undirected edges $(i, j)$ in $\mathcal{G}$ if there exists a set $S \subseteq V \setminus \{X_i, X_j\}$ such
   that $X_i$ and $X_j$ are conditionally independent given $S$, $X_i \perp\!\!\!\perp X_j \mid S$
3: orient all *immoralities* in the remaining graph structure $\mathcal{G}$
4: while we can orient edges in $\mathcal{G}$ do
5:     orient all edges if the reverse direction would introduce an immorality
6:     orient all edges if the reverse direction would introduce a directed cycle
7: end while
8: return the maximally oriented CPDAG $\mathcal{G}$
```

Example. To see how constraint-based approaches work, let us consider the following example on five variables, inspired by [55]. The ground truth has been visualized in Figure 3.3.

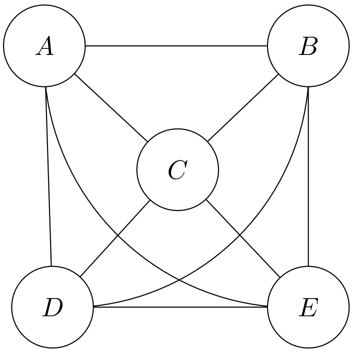

Figure 3.1: Complete undirected graph.

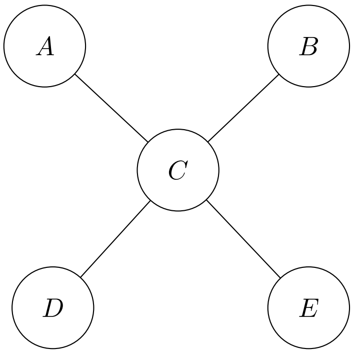

Figure 3.2: Skeleton inferred by conditional independence tests.

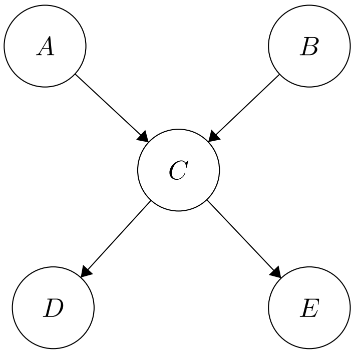

Figure 3.3: Ground truth, also recovered using a constraint-based approach.

To obtain the correct graph using a constraint-based approach, we will first learn the *skeleton* of the structure. The skeleton consists of all edges that are dependent, even when we condition on all other variables. To identify such a skeleton, we will start with a fully connected graph in Figure 3.1 and keep on removing edges $X_i - X_j$ if we can find a set S such that $X_i \perp\!\!\!\perp X_j \mid S$. That is, we are looking for a set such that X_i and X_j are independent when we condition on the variables in the set.

Only four direct dependencies $A-C$, $B-C$, $C-D$, and $C-E$ are included in the ground truth in Figure 3.3. Firstly, we can remove the edge $A-B$, as A and B are conditionally independent, even given the empty set. Additionally, all edges from either A or B to either D or E can be removed, as all four pairs of variables are conditionally independent given C. Lastly, we know that D and E are independent given C as well. Therefore, we see that we can successfully recover the skeleton which has been shown in Figure 3.2.

Now, to orient the edges, we will first see if we can detect any immoralities. Consider the triple $A-C-B$. We have verified that there is no edge between A and B. Furthermore, we did not require conditioning on C to ensure that A and B were conditionally independent. Therefore, the triplet $A-C-B$ is considered an immorality, and we can orient these edges as $A \rightarrow C \leftarrow B$. Subsequently, we can orient the other two edges as well. If we assumed orientation $C \rightarrow D$, then we would have introduced an additional immorality $A \rightarrow C \leftarrow D$. Therefore, according to the Meek rules, the proper orientations are $C \rightarrow D$ and $C \rightarrow E$, which indeed corresponds to the ground truth.

PCMCI: Extending to time series data. The SGS/IC and PC algorithm were all developed for time-independent data, and were therefore only able to model instantaneous relations. The PC-algorithm can also be extended to time series data. However, the autocorrelations existing in time series data lead to high false positive rates for the conditional independence tests, and the PC-algorithm can therefore not be used directly [69].

To overcome this issue, the PCMCI algorithm [69] was introduced for time series data. First, the PC-algorithm is used to remove edges by checking for conditional independence between time-lagged variables, although this may lead to a large number of false positives due to autocorrelation. The remaining edges are then checked using Momentary Conditional Independence (MCI) tests to (hopefully) address these false positives. These MCI tests are used to verify whether a time lagged-variable $X^i_{t-\tau}$ and a non time-lagged variable X^j_t are independent when we condition on both parent sets identified by the PC algorithm without the time-lagged variable $X^j_{t-\tau}$,

$$\text{MCI} : X^i_{t-\tau} \perp\!\!\!\perp X^j_t \mid \text{Pa}\left(X^i_{t-\tau}\right), \text{Pa}\left(X^j_t\right) \setminus X^j_{t-\tau}. \tag{3.4}$$

Note that here, the subscript represents the time index and the superscript represents the variable index. The orientation of edges is done using the same rule set as the PC-algorithm, but as the future cannot cause the past, we know that all edges should be oriented from past to future, so $X^i_t \rightarrow X^j_{t'}$ if $t < t'$. The initial PCMCI algorithm was unable to detect instantaneous causal relations, but thishas been made possible by extending the algorithm in [68].

Several other constraint-based algorithms exists, mostly focused on improving the computational performance of the PC-algorithm. The Fast Causal Inference (FCI) algorithm was proposed in 1993 [74] and the Really Fast Causal Inference (RFCI) algorithm was proposed in 2012 [17]. The FCI algorithm enjoys several time series extensions in the form of the SVAR-FCI algorithm [50] and the tsFCI algorithm [25].

3.2 Noise Structure Based Approaches

As mentioned before, it is impossible to determine the directionality of the effect without certain noise assumptions. In other words, the corresponding structure is then *unidentifiable*. Let us see explain this phenomenon in the following example.

Example of unidentifiability of a linear bivariate Gaussian model. Consider a linear Gaussian SEM on two variables which is defined as follows:

$$X = \varepsilon_X, \qquad \varepsilon_X \sim \mathcal{N}(0,1). \tag{3.5}$$

$$Y = 0.8X + \varepsilon_Y, \qquad \varepsilon_Y \sim \mathcal{N}(0,0.6). \tag{3.6}$$

These values have been chosen such that both X and Y have a mean of zero and a variance of one.

From Equation 3.6, we can deduce that $X \to Y$. However, when we purely look at observational data, we cannot distinguish between the model from defined by Equations 3.5 and 3.6 and the model given by

$$X = 0.8Y + \tilde{\varepsilon}_X, \qquad \tilde{\varepsilon}_X \sim \mathcal{N}(0, 0.6) \tag{3.7}$$

$$Y = \tilde{\varepsilon}_Y, \qquad \tilde{\varepsilon}_Y \sim \mathcal{N}(0, 1) \tag{3.8}$$

We can show that in both scenarios, X, Y are both standard normal random variables, where their joint distribution is a bivariate normal distribution,

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 1.0 \end{bmatrix} \right). \tag{3.9}$$

Therefore, both models result in the same joint distribution of X and Y. This has also been shown in Figure 3.4 and Figure 3.5, where we have visualized the joint distributions of both models.

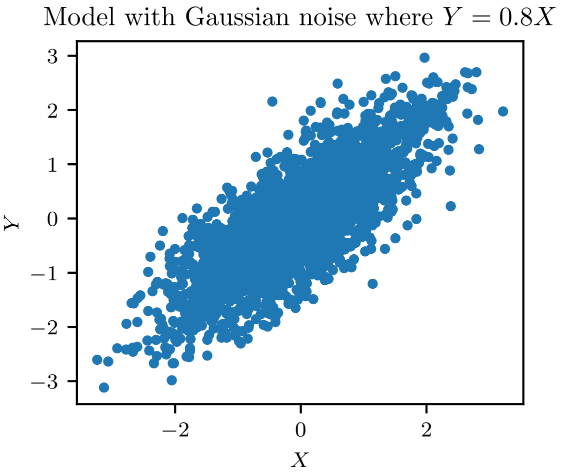

Figure 3.4: Scatter plot of the linear Gaussian model defined by Equations 3.6 and 3.5.

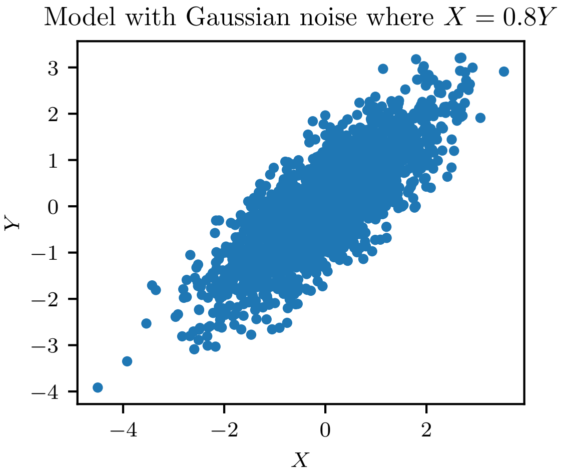

Figure 3.5: Scatter plot of the linear Gaussian model defined by Equations 3.7 and 3.8.

As both linear Gaussian models have the same joint distribution, the "correct" model is *unidentifiable*. The general statement that such a linear Gaussian model is *unidentifiable* has been shown in [72].

Example of identifiability of a linear model with uniform noise. However, the directionality is detectable as soon as we make some assumptions on the noise variables $\varepsilon, \tilde{\varepsilon}$. Let us see how we can discover the correct direction in the following example.

Let us consider the scenario where our noise components follow a *non-Gaussian* distribution such as the uniform distribution. The noise residuals are now uniformly distributed in such a way that the mean and variance of X and Y do not change,

$$X = \varepsilon_X, \qquad \varepsilon_X \sim U\left(-\sqrt{3}, \sqrt{3}\right). \tag{3.10}$$

$$Y = 0.8X + \varepsilon_Y, \qquad \varepsilon_Y \sim U\left(-0.6\sqrt{3}, 0.6\sqrt{3}\right).. \tag{3.11}$$

The scatter plots have been given in Figure 3.6 and Figure 3.7.

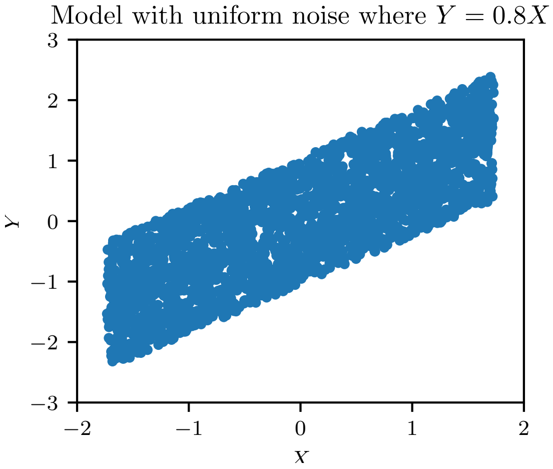

Figure 3.6: Scatter plot of the linear model $Y = 0.8X$ with uniform noise.

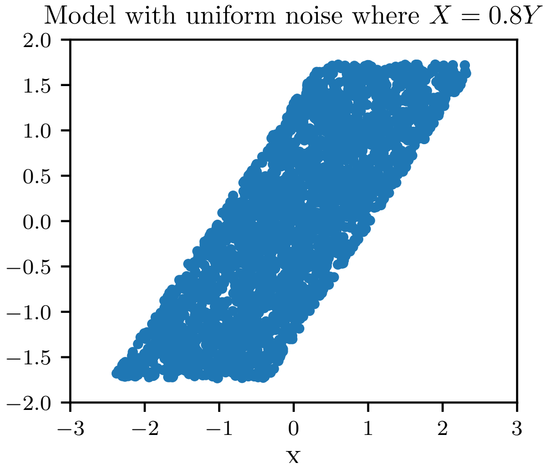

Figure 3.7: Scatter plot of the linear model $X = 0.8Y$ with uniform noise.

Both scatter plots depicted show the linear relationship between X and Y. Nevertheless, we see that there is a clear difference between Figure 3.6 and Figure 3.7. Figure 3.7 seems to show a steeper linear relationship between X and Y. Therefore, the two models now seem to exhibit a different *joint distribution* of X and Y.

Now that we have shown there is indeed a difference in joint distribution between these two models, let us consider how we can recover the *correct* direction of influence. For this, we will assume that the correct direction is $X \to Y$, and the additive noises ε_X and ε_Y are independent uniform random variables as defined by Equations 3.10 and 3.11. To verify the correct direction, we can regress Y on X, yielding approximately the regression line $Y = 0.8X$. Furthermore, we can also regress X on Y, yielding the regression line $X = 0.8Y$. Note that regression coefficients are the same because the variance parameters were carefully selected. These regression lines have been visualized in Figure 3.8. We see that the black regression line in the correct direction sits nicely in the middle of the scatter plot. However, the red regression line for $Y \to X$ seems to low for small values of X, and too high for large values of X. This can be further visualized by looking at the *residuals* of the corresponding regressions, which have been plotted in Figure 3.9 and Figure 3.10.

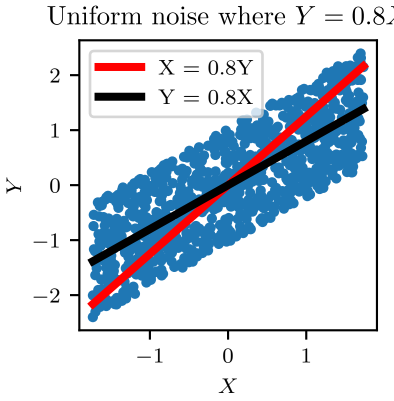

Figure 3.8: Scatter plot of the linear model $Y = 0.8X$ with uniform noise. Both regression lines $X \to Y$ (black) and $Y \to X$ (red) have been plotted as well.

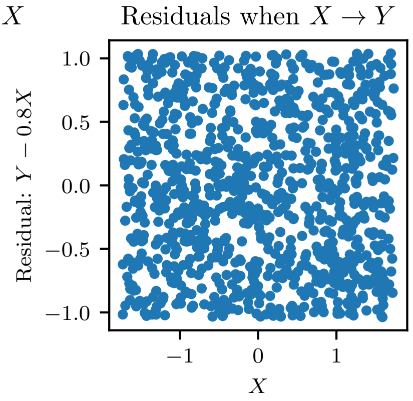

Figure 3.9: Scatter plot of the residuals of the linear model $X = 0.8Y$ with uniform noise, corresponding to the distance to the black line in Figure 3.8.

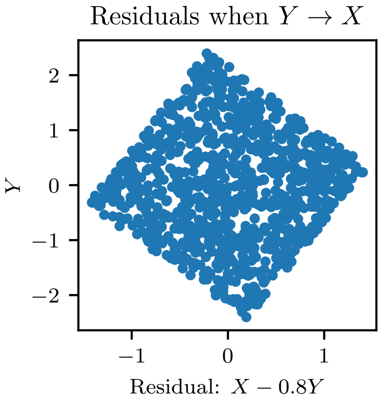

Figure 3.10: Scatter plot of the residuals of the linear model $Y = 0.8X$ with uniform noise, corresponding to the distance to the red line in Figure 3.8.

We see that the residuals in the correct direction $X \to Y$ are independent of X and Y, whereas the residuals in the incorrect direction $Y \to X$ are *dependent* on X and Y. As we assumed the residuals to be independent of X and Y, the only plausible direction is indeed $X \to Y$.

LiNGAM. Assuming that our residuals follow some non-Gaussian distribution, we are able to determine whether $X \to Y$ or $X \leftarrow Y$ by verifying whether the residuals are independent. This phenomenon is utilized in the *Linear Non-Gaussian Acyclic Model* (LiNGAM) [72].

LiNGAM assume a linear non-Gaussian model with instantaneous interactions,

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}, \tag{3.12}$$

where $\mathbf{x}$ is their notation for a data matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$. Equation 3.12 can also be rewritten as

$$\mathbf{x} = \mathbf{A}\mathbf{e}, \tag{3.13}$$

where $\mathbf{A} = (\mathbf{I} - \mathbf{B})^{-1}$. The first step in LiNGAM is to perform an independent component analysis (ICA) to obtain an estimate $\tilde{\mathbf{W}}$ of $\mathbf{W} = \mathbf{A}^{-1}$. This $\mathbf{W}$ needs to be permuted and normalized in order to obtain an estimate $\hat{\mathbf{B}}$ of $\mathbf{B}$. Lastly, a causal ordering can be derived from this matrix $\tilde{\mathbf{B}}$ which yields a suitable estimate for $\mathbf{B}$.

VARLiNGAM: Extending to time series data. We can extend the LiNGAM setting to a time-series setting using the VARLiNGAM approach [39]. Instead of only instantaneous relations, we also model time-lagged relationships through a VAR(τ_{max}) model,

$$\mathbf{x}_t = \sum_{i=0}^{\tau_{max}} \mathbf{A}_i \mathbf{x}_{t-i} + \mathbf{e}. \tag{3.14}$$

This model can also be rewritten without an instantaneous effect as

$$\mathbf{x}_t = \sum_{i=1}^{\tau_{max}} \mathbf{M}_i \mathbf{x}_{t-i} + \mathbf{e}, \tag{3.15}$$

where $\mathbf{A}_i = (I - \mathbf{A}_0)\,\mathbf{M}_i$.

We can fit such a VAR(τ_{max}) model on our data $\mathbf{x}$, which yields residuals $\mathbf{e}$. On these residuals, we can can do a LiNGAM analysis as mentioned in the paragraph above, which yields an instantaneous coefficient matrix for the instantaneous relations $\hat{\mathbf{A}}_0$. From $\hat{\mathbf{A}}_0$ and $\hat{\mathbf{M}}_i$, in turn, we can compute $\hat{\mathbf{A}}_i = \left(I - \hat{\mathbf{A}}_0\right)\hat{\mathbf{M}}_i$. Using this VARLiNGAM approach, we can estimate the instantaneous effects while taking information from past values into account.

3.3 Score-Based Structure Learning

Score-based approaches assess the validity of a given structure by assigning a score to each structure. This score is determined by a scoring function $S(\mathbf{X}, \mathcal{G})$. Consequently, the optimal structure is the structure $\mathcal{G}^*$ that maximizes the scoring function $S(\mathbf{X}, \mathcal{G})$, while remaining acyclic,

$$\mathcal{G}^* = \underset{\text{structures } \mathcal{G}}{\arg\max}\; S(\mathbf{X}, \mathcal{G}) \text{ such that } \mathcal{G} \in \texttt{DAGs}. \tag{3.16}$$

Scoring functions that are often used in the literature are the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC), which are both based on the likelihood function of the data.

Globally Optimal Bayesian Network learning using Integer Linear Programming

Finding the structure $\mathcal{G}^*$ that maximizes the scoring function $S(\mathbf{X}, \mathcal{G})$ has been shown to be NP-hard and therefore we cannot expect to solve this efficiently for more than a couple of dozens of nodes. Nevertheless, several approaches exists that can quite efficiently find the optimal structure given a scoring function $S(\mathbf{X}, \mathcal{G})$. One of the most well-known exact solvers is known as GOBNILP [19], which is short for *Globally Optimal Bayesian Network learning using Integer Linear Programming.*

Its key to success relies on rewriting the score-based approach. Instead of directly maximizing a scoring function $S(\mathbf{X}, \mathcal{G})$ such that $\mathcal{G}$ is acyclic, they rewrite the problem in Equation 3.16 to a binary Integer Linear Program (ILP). A binary ILP consists of an objective function to maximize subject to a set of constraints, where the assignment variables can only attain the value zero or one, hence the name Integer. By defining the assignment variables and constraints in a smart way, we can create an ILP such that the optimal value of the ILP corresponds to the optimal value of the problem in Equation 3.16

The Integer Linear Program. The Integer Linear Program has been given below. An explanation of the objective function in Equation 3.17 and the sets of constraints in Equations 3.18 - 3.20 follow afterwards.

$$\text{maximize} \quad \sum_{i \in V, J \in \mathcal{P}(i)} c_{i \leftarrow J} x_{i \leftarrow J} \tag{3.17}$$

$$\text{subjsect to} \quad \sum_{J \in \mathcal{P}(i)} x_{i \leftarrow J} = 1 \qquad \forall\, i \in V \tag{3.18}$$

$$\sum_{i \in C} \sum_{J \in \mathcal{P}(i): J \cap C = \emptyset} x_{i \leftarrow J} \geq 1 \qquad \forall\, C \subseteq V, \tag{3.19}$$

$$x_{i \leftarrow J} \in \{0, 1\} \qquad \forall\, i \in V, J \in \mathcal{P}(i) \tag{3.20}$$

The authors first define *binary variables* $x_{i \leftarrow J}$ for each variable $i \in V$, where V corresponds to the set of variables, and each possible parent set $J \in \mathcal{P}(i)$. Here, $\mathcal{P}(j)$ corresponds to all possible parent sets of j, which is equal to the power set of $V \setminus \{i\}$. As each variable has $2^{|V|-1}$ different parent sets, we have an exponential number of binary variables $x_{i \leftarrow J}$. Note that these variables can only attain a value of either zero or one, as is shown in Equation 3.20. If $x_{i \leftarrow J} = 0$, then J is not the exact parent set of variable i. If $x_{i \leftarrow J} = 1$, then J corresponds to the parent set of variable i. Now, $c_{i \leftarrow J}$ corresponds to the scoring function when we use the variables in the parent set J to predict the values of variable i.

Continuing with the constraints, we know that each variable i must have exactly one parent set. This set of constraints correspond to Equation 3.18. Summing over all $0 - 1$ variables that representing all $\mathcal{P}(i)$ possible parent sets of variable i should yield exactly the value 1, indicating that exactly one parent set has been assigned to each variable.

The second set of constraints in Equation 3.19 ensures that there are no cycles in the structure. For example, variable 1 cannot be in the parent set of variable 2 when variable 2 is already in the parent set of variable 1. This constraint has been cleverly written in Equation 3.19. Every subset $C \subseteq V$ of the variables must contain at least one vertex who has no parent in that subset. To see why this is true, if we would find a subset $C \subseteq V$ such that all vertices have a parent in the subset, then there must exists a cycle.

Solving the Integer Linear Program. A great advantage of casting this score-based learning problem to an ILP is that there is no need to reinvent any new solver for the problem. Nowadays, highly optimized off-the-shelf ILP solvers exist thanks to decades of research into ILP solvers [5]. However, as there are still an exponential number of assignment variables $x_{i \leftarrow J}$, and the number of constraints is exponential as well, especially the constraints in Equation 3.19, an optimal solution may still take a long time to acquire.

Note that GOBNILP employs many additional tricks to speed up the computations which are not covered here. Furthermore, many improvements have been proposed to increase the computational performance of GOBNILP. In fact, the GOBNILP algorithm is actively maintained and new versions are continuously developed to include the newest performance enhancements, such as different formulations of the constraints [78] and a branch-and-cut approach to reduce the search space [4]. Furthermore, a Python version `pygobnilp` has been introduced as well.

NOTEARS

NOTEARS (*Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning*) is a method developed by Zheng et al. [91]. Rather than trying to find an optimal acyclic structure, they settle for a suboptimal structure with the benefit that their method can scale up to approximately one hundred nodes.

They authors assume a linear Structural Equation Model of the form

$$X = W^T X + z, \tag{3.21}$$

where $X = (X_1, \ldots X_p) \in \mathbb{R}^p$ is defined a random vector and $z = (z_1, \ldots, z_p)$ defines a random noise vector. Their goal is to infer the matrix $W \in \mathbb{R}^{p \times p}$ from n independent samples of X, forming a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The most difficult hurdle to overcome is the combinatorial constraint that the inferred structure $G(W)$ must be a directed acyclic graph. In their paper, they present a novel strategy to enforcing acyclicity of the structure. Rather than solving the (partly) combinatorial optimization problem in the left-hand side of Equation 3.30, they translate it to an equivalent continuous optimization problem in the right-hand side of Equation 3.30. For any continuous scoring function $F(W)$, they show the equivalence

$$\begin{array}{ll} \min\limits_{W \in \mathbb{R}^{p \times p}} & F(W) \\ \text{subject to} & G(W) \in \mathsf{DAGs} \end{array} \iff \begin{array}{ll} \min\limits_{W \in \mathbb{R}^{p \times p}} & F(W) \\ \text{subject to} & h(W) = 0, \end{array} \tag{3.22}$$

for some function $h(\cdot)$ that enforces acyclicity such that $h(W) = 0$ if and only if $G(W)$ is a directed acyclic graph. Now, the function $F(\cdot)$ that the authors attempt to optimize is the least-squares loss plus an ℓ_1 regularization on the weighted adjacency matrix to encourage sparsity. In mathematical notation, their scoring functions equals

$$F(W) = \frac{1}{2n} \|\mathbf{X} - \mathbf{X}W\|_F^2 + \lambda \|W\|_1, \tag{3.23}$$

where $\mathbf{X} \in \mathbb{R}^{n \times p}$ represent a data matrix of n samples that are p-dimensional. The continuous function $h(W)$ that enforces acyclicity corresponds to

$$h(W) = \operatorname{Tr}\left(e^{(W \circ W)}\right) - p = \operatorname{Tr}\left(\sum_{k=1}^{\infty} \frac{(W \circ W)^k}{k!}\right). \tag{3.24}$$

Here, $\operatorname{Tr}(\cdot)$ represents the trace operator which sums all diagonal entries of its argument, and $W \circ W$ represents the *Hadamard product* of two matrices. In other words, it is the element-wise square of the matrix W.

To explain the function h in greater detail, let us first remark that the trace of $(W \circ W)^k$ corresponds to a weighted sum of all cycles of length k in the matrix $W \circ W$. Therefore, the matrix $W \circ W$ does not contain any cycles of length k if and only if $(W \circ W)^k = 0$. Therefore, $W \circ W$ does not contain any cycles if and only if $h(W) = 0$.

Optimization. The authors propose the *augmented Lagrangian* method to solve the right-hand side of Equation 3.30, which now contains an augmented quadratic penalty,

$$\min_{W \in \mathbb{R}^{p \times p}} F(W) + \frac{\rho}{2} |h(W)|^2 \text{ subject to } h(W) = 0, \tag{3.25}$$

where ρ represents the penalty parameter that is initially equal to one but is iteratively increased.

To adhere to the constraint that $h(W) = 0$, a Lagrange multiplier α is included, yielding the augmented Lagrangian

$$L^\rho(W, \alpha) = F(W) + \frac{\rho}{2} |h(W)|^2 + \alpha h(W). \tag{3.26}$$

To find a stationary point of the augmented Lagrangian, an iterative dual ascent procedure is proposed. Initially, the penalty parameter ρ_0 is equal to one and the Lagrange multipliers α_0 is equal to zero. For these values, a local optimum W_1 is found using the L-BFGS-B optimization method [95]. If this optimization has provided enough progress with respect to the acyclicity constraint, that is, $h(W_1) < ch(W_0)$ for some hyperparameter $c \in (0,1)$, then we increase the Lagrange multiplier α and again use L-BFGS-B to find a coefficient matrix W_2. If not, we keep multiplying ρ by ten and finding a new local optima W_1 until $h(W_1) < ch(W_0)$. When a suitable local optimum W_1 has been found, the Lagrange multiplier α_0 ins increased by $\rho h(W_1)$.

Continuing this process, the penalty parameter ρ and Lagrange multiplier α_t keep increasing and local optima are recovered such that $h(W_t)$ is closer and closer to zero, After several iterations, we have found a local optimum to Equation 3.25 such that $h(W)$ is sufficiently close to zero, say no more than some value ϵ.

As a final procedure, the solution to the equality constrained program (ECP), W_{ECP} will be thresholded. This means that all entries of W_{ECP} that are smaller in absolute value than some threshold ω will be set to zero to reduce the number of false positives [94]. Experiments done by the authors demonstrate that thresholding increases the accuracy in structure learning, and in their experiments a threshold value of $\omega = 0.30$ has been proposed. The pseudocode for the NOTEARS procedure has been given in Algorithm 3.2.

Algorithm 3.2: NOTEARS

Input: initial guess W_0, progress rate $c \in (0,1)$, tolerance $\epsilon > 0$, threshold $\omega > 0$.
Output: an acyclic coefficient matrix $\widehat{W} \in \mathbb{R}^{p \times p}$.

```
for $t = 0, 1, 2, \ldots$ do:
    solve primal $W_{t+1} \leftarrow \arg\min_W L^\rho(W, \alpha_t)$ with $\rho$ such that $h(W_{t+1}) < ch(W_t)$.
    dual ascent $\alpha_{t+1} \leftarrow \alpha_t + \rho h(W_{t+1})$.
    if $h(W_{t+1}) < \epsilon$, set $W_{\mathsf{ECP}} = W_{t+1}$ and break.
end for
return the thresholded matrix $\widehat{W} := W_{\mathsf{ECP}} \circ 1(|W_{\mathsf{ECP}}| > \omega)$.
```

Note that only a local minimum is guaranteed, and not a global minimum, as the search space is non-convex. Given that the problem at hand is NP-hard, this comes as no surprise. However, the fact that only a local minimum is guaranteed does not seem to be a detrimental problem. The authors have compared the local optimum found using Algorithm 3.30 to the global optimum found using an exact program. In this scenario, the authors have used the GOBNILP [19] program to find the global minimum. Quite often, the local optimum was close to the global optimum, which demonstrates that the non-convexity is not a large issue in practice.

DYNOTEARS: Extending to time series data. We have seen that NOTEARS only allows for *instantaneous* relationships. However, this can easily be extended to also allow for *time-lagged* relationships. DYNOTEARS [58] assumes that, apart from instantaneous relations, there are also linear time-lagged relations.

The extension of DYNOTEARS to NOTEARS is similar to the extension of VARLiNAM to LiNGAM, which was discussed in Section 3.2. Instead of assuming the instantaneous linear model

$$X = W^T X + z, \tag{3.27}$$

we now also model time-lagged linear relations, up to a time lag of τ_{max},

$$X_{t,\cdot} = X_{t,\cdot} W + \sum_{i=1}^{\tau_{max}} A_i X_{t-i,\cdot} + z, \tag{3.28}$$

where z again represents zero-mean random noise.

Instead of minimizing the cost function $F(W)$ of Equation 3.23 subject to $h(W) = 0$, we now minimize the cost function

$$F'(W) = \frac{1}{2n} \sum_{t=\tau_{max}+1}^{T} \left\| X_{t,\cdot} - X_{t,\cdot}W - \sum_{i=1}^{\tau_{max}} A_i X_{t-i,\cdot} \right\|_2^2 + \lambda_1 \|W\|_1 + \lambda_2 \sum_{i=1}^{\tau_{max}} \|A_i\|_1 , \tag{3.29}$$

where W captures the instantaneous relations, and A_i captures the relations that are time-lagged by i time steps. Additionally, We see that we have added the time lagged linear relations to the model, as well as an extra sparsity penalty to the autoregressive coefficient matrices A_i. Furthermore, n here corresponds to the effective sample size, for which we have $n = p(T - \tau_{max})$. In their paper, they have $n = p(T + 1 - \tau_{max})$, but their time series also start at time index $t = 0$ rather than our convention of $t = 1$, so therefore we have this small discrepancy of notation.

Just as in NOTEARS, we can enforce acyclicity using a continuous optimization approach by using the translated problem,

$$\begin{array}{rl} \min\limits_{W \in \mathbb{R}^{p \times p}} & F'(W) \\ \text{subject to} & G(W) \in \mathsf{DAGs} \end{array} \iff \begin{array}{rl} \min\limits_{W \in \mathbb{R}^{p \times p}} & F'(W) \\ \text{subject to} & h(W) = 0, \end{array} \tag{3.30}$$

Note that DYNOTEARS does not enforce acyclicity on the autoregressive coefficient matrices A_i. However, this would certainly be possible if we would change the acyclicity constraint function $h(\cdot)$.

Other TEARS. The interesting approach first introduced in NOTEARS has sparked a great interest in such methods, and dozens of papers that mimic or improve on the continuous optimization constraint of NOTEARS have been introduced. As mentioned, DYNOTEARS extends upon NOTEARS by allowing linear time-lagged relationships. Addtionally, there are several other algorithm that improve upon NOTEARS.

Firstly, researchers have sought for better functions $h(\cdot)$ that enforce acyclicity. The authors of NO FEARS [87] analyze the acyclicity constraint function $h(\cdot)$ used in NOTEARS and conclude that their method of solving Equation 3.30 using the augmented Lagrangian method does not always convergence to a feasible solution. In other words, the solution may not satisfy the constraint $h(W) = 0$. They propose another similar function that enforces acyclicity and they show that their newly proposed method is an improvement over NOTEARS.

Secondly, the authors of NO-BEARS [47] also employ a different approach to verify acyclicity. Instead of investigating the trace exponential of $W \circ W$, the authors of NO BEARS use the spectral radius of $W \circ W$ to enforce acyclicity. Furthermore, they allow for polynomial relations between variables instead of a linear relationship to allow for more complex models.

Lastly, the authors of NOTEARS propose an extension to their own approach. Instead of modeling linear relationships, they propose a neural network such that non-linear relationships can be modeled [93]. A similar approach is also employed in [89], but they use a variational autoencoder to model non-linear relations and there acyclicity constraint is formulated slightly different.

Bibliography

[1] *Vector Autoregressive Models for Multivariate Time Series*, pages 385–429. Springer New York, New York, NY, 2006. 55

[2] Awad H. Al-Mohy and Nicholas J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2010. 75, 94

[3] M. Andrle, L. Rebollo-Neira, and E. Sagianos. Backward-optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 11(9):705–708, 2004. 92

[4] Mark Barlett and James Cussens. Advances in bayesian network learning using integer programming. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 182–191, Arlington, Virginia, USA, 2013. AUAI Press. 22

[5] Mark Bartlett and James Cussens. Integer linear programming for the bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017. Combining Constraint Solving with Mining and Learning. 22

[6] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics Data Analysis*, 120:70–83, 2018. 105

[7] Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946. 52

[8] Thomas Blumensath and Mike Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. 03 2007. 77

[9] Graham Brightwell and Peter Winkler. Counting linear extensions is #p-complete. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, page 175–181, New York, NY, USA, 1991. Association for Computing Machinery. 33

[10] Prabir Burman, Edmond Chow, and Deborah Nolan. A cross-validatory method for dependent data. *Biometrika*, 81:351–358, 1994. 108

[11] T. Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011. 134

[12] Nancy Cartwright. Are rcts the gold standard? *BioSocieties*, 2(1):11–20, 2007. 4

[13] Rui Castro and Robert Nowak. Likelihood based hierarchical clustering and network topology identification. In Anand Rangarajan, Mário Figueiredo, and Josiane Zerubia, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 113–129, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 40

[14] S. CHEN, S. A. BILLINGS, and W. LUO. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989. 77

[15] Wenyu Chen, Mathias Drton, and Y Samuel Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4):973–980, 09 2019. 134

[16] David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*, pages 121–130. Springer New York, New York, NY, 1996. 15

[17] Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 2012. 18

[18] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990. 4

[19] James Cussens. Bayesian network learning with cutting planes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 153–160, Arlington, Virginia, USA, 2011. AUAI Press. 21, 24

[20] Aramayis Dallakyan and Mohsen Pourahmadi. Learning bayesian networks through birkhoff polytope: A relaxation method. *CoRR*, abs/2107.01658, 2021. 64

[21] Ivan Damnjanovic, Matthew E. P. Davies, and Mark D. Plumbley. Smallbox - an evaluation framework for sparse representations and dictionary learning algorithms. In Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent, editors, *Latent Variable Analysis and Signal Separation*, pages 418–425, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 87

[22] Tristan Deleu, António Góis, Chris Chinenye Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 128

[23] Vera Djordjilović, Monica Chiogna, and Jiří Vomlel. An empirical comparison of popular structure learning algorithms with a view to gene network inference. *International Journal of Approximate Reasoning*, 88:602–613, 2017. 117

[24] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. 98

[25] Doris Entner and Patrik Hoyer. On causal discovery from time series data using fci. *Proceedings of the 5th European Workshop on Probabilistic Graphical Models, PGM 2010*, 09 2010. 18

[26] Kim Esbensen and Paul Geladi. Principles of proper validation: use and abuse of re-sampling for validation. *Journal of Chemometrics*, 24:168 – 187, 03 2010. 105

[27] Ronald Aylmer Fisher et al. 014: On the" probable error" of a coefficient of correlation deduced from a small sample. 1921. 16

[28] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman amp; Co., USA, 1990. 77

[29] Maxime Gasse, Alex Aussem, and Haytham Elghazel. An experimental comparison of hybrid algorithms for bayesian network structure learning. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 58–73, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 117

[30] Sarah Gelper, Ines Wilms, and Christophe Croux. Identifying demand effects in a large network of product categories. *Journal of Retailing*, 92(1):25–39, 2016. 124

[31] M. Gharavi-Alkhansari and T.S. Huang. A fast orthogonal matching pursuit algorithm. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 3, pages 1389–1392 vol.3, 1998. 77

[32] Hemant S. Goklani, Jignesh N. Sarvaiya, and A. M. Fahad. Image reconstruction using orthogonal matching pursuit (omp) algorithm. In *2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking*, pages 1–5, 2014. 78

[33] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969. 5

[34] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001. 73

[35] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 40

[36] Sander Hofman. Making euv: From lab to fab, Mar 2022. 3

[37] Guoxian Huang and Lei Wang. High-speed signal reconstruction with orthogonal matching pursuit via matrix inversion bypass. pages 191–196, 10 2012. 87

[38] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 2nd edition, 2018. 27

[39] Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(56):1709–1731, 2010. 21

[40] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975. 93

[41] Hidde De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *JOURNAL OF COMPUTATIONAL BIOLOGY*, 9:67–103, 2002. 3

[42] A. B. Kahn. Topological sorting of large networks. *Commun. ACM*, 5(11):558–562, nov 1962. 76

[43] Wagner A. Kamakura and Wooseong Kang. Chain-wide and store-level analysis for cross-category management. *Journal of Retailing*, 83(2):159–170, 2007. 124

[44] Mahdi Khosravy, Nilanjan Dey, and Carlos Duque. *Compressive Sensing in Health Care*. 10 2019. 77

[45] S. N. Lahiri. *Bootstrap Methods*, pages 17–43. Springer New York, New York, NY, 2003. 98

[46] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988. 4

[47] Hao-Chih Lee, Matteo Danieletto, Riccardo Miotto, Sarah Cherng, and Joel T. Dudley. Scaling structural learning with no-bears to infer causal transcriptome networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 25:391–402, 2020. 25

[48] Hanxi Li, Yongsheng Gao, and Jun Sun. Fast kernel sparse representation. In *2011 International Conference on Digital Image Computing: Techniques and Applications*, pages 72–77, 2011. 87

[49] Rami Mahdi and Jason Mezey. Sub-local constraint-based learning of bayesian networks using a joint dependence criterion. *Journal of Machine Learning Research*, 14(13):1563–1603, 2013. 16

[50] Daniel Malinsky and Peter Spirtes. Causal structure learning from multivariate time series in settings with unmeasured confounding. In *Proceedings of 2018 ACM SIGKDD workshop on causal discovery*, pages 23–47. PMLR, 2018. 18

[51] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 77

[52] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. 16

[53] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246 – 270, 2009. 135

[54] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. , 21(6):1087–1092, June 1953. 40

[55] Brady Neal. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes*, 2020. 16, 17

[56] William B. Nicholson, David S. Matteson, and Jacob Bien. Varx-l: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 33(3):627–651, 2017. 133

[57] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2022. Published electronically at `https://oeis.org/A003024`. 135

[58] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1595–1605. PMLR, 26–28 Aug 2020. 24

[59] Koen Pauwels. How retailer and competitor decisions drive the long-term effectiveness of manufacturer promotions for fast moving consumer goods. *Journal of Retailing*, 83(3):297–308, 2007. 124

[60] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986. 4, 16

[61] Judea Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, 2000. 17

[62] Judea Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, USA, 2nd edition, 2009. v, 1, 2, 4

[63] Judea Pearl and Thomas Verma. A theory of inferred causation. In *KR*, 1991. 15

[64] J. PETERS and P. BÜHLMANN. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014. 134

[65] K. B. Petersen and M. S. Pedersen. The matrix cookbook, October 2008. Version 20081110. 58

[66] Joseph Ramsey and Bryan Andrews. Fask with interventional knowledge recovers edges from the sachs model. *ArXiv*, abs/1805.03108, 2018. 128

[67] R. W. Robinson. Counting unlabeled acyclic digraphs. In Charles H. C. Little, editor, *Combinatorial Mathematics V*, pages 28–43, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg. 32

[68] Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on Uncertainty in Artificial Intelligence*, pages 1388–1397. PMLR, 2020. 18

[69] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996, 2019. 18

[70] Karen Sachs, Omar Perez, Dana Pe'er, Douglas Lauffenburger, and Garry Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science (New York, N.Y.)*, 308:523–9, 05 2005. vii, vii, ix, 128, 129, 131

[71] Rajen Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *Annals of Statistics*, 48, 04 2018. 16

[72] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyv228;rinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006. 19, 21, 88

[73] Konstantinos Skianis, Nikolaos Tziortziotis, and Michalis Vazirgiannis. Orthogonal matching pursuit for text classification. *ArXiv*, abs/1807.04715, 2018. 78

[74] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT press, 2nd edition, 2000. 17, 18

[75] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991. 17

[76] Shuba Srinivasan, Koen Pauwels, Dominique M. Hanssens, and Marnik G. Dekimpe. Do promotions benefit manufacturers, retailers, or both? *Management Science*, 50(5):617 – 629, 2004. Cited by: 177; All Open Access, Green Open Access. 124

[77] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974. 105

[78] Milan Studený and James Cussens. Towards using the chordal graph polytope in learning decomposable models. *International Journal of Approximate Reasoning*, 88:259–281, 2017. 22

[79] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 68

[80] Ryan Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39, 05 2010. 70

[81] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004. 134

[82] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 87

[83] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 10 2006. 117

[84] Alexander L. Tulupyev and Sergey I. Nikolenko. Directed cycles in bayesian belief networks: Probabilistic semantics and consistency checking complexity. In Alexander Gelbukh, Álvaro de Albornoz, and Hugo Terashima-Marín, editors, *MICAI 2005: Advances in Artificial Intelligence*, pages 214–223, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 6

[85] John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2(0):5–12, 1953. 52

[86] Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Comput. Surv.*, mar 2022. Just Accepted. 15, 117

[87] Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3895–3906. Curran Associates, Inc., 2020. 25

[88] Norbert Wiener. The theory of prediction. *Modern mathematics for engineers*, 1956. 5

[89] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7154–7163. PMLR, 09–15 Jun 2019. 25, 128, 134

[90] Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10(19):555–568, 2009. 77, 134

[91] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018. 23

[92] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc. 51, 64, 126, 128, 130

[93] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020. 25, 134

[94] Shuheng Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. 24

[95] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997. 24

[96] Hufei Zhu, Wen Chen, and Yanpeng Wu. Efficient implementations for orthogonal matching pursuit. *Electronics*, 9:1507, 09 2020. 87