Identifiability of Linear AMP Chain Graph Models

Yuhao Wang and Arnab Bhattacharyya

National University of Singapore
yuhaowang@u.nus.edu, arnabb@nus.edu.sg

Abstract

We study identifiability of linear Andersson-Madigan-Perlman (AMP) chain graph models, which are a common generalization of linear structural equation models and Gaussian graphical models. AMP models are described by DAGs on chain components which themselves are undirected graphs. For a known chain component decomposition, we show that the DAG on the chain components is identifiable if the determinants of the residual covariance matrices of the chain components are equal (or more generally, monotone non-decreasing in topological order). This condition extends the equal variance identifiability criterion for Bayes nets, and it can be generalized from determinants to any super-additive function on positive semidefinite matrices. When the component decomposition is unknown, we describe conditions that allow recovery of the full structure using a polynomial time algorithm based on submodular function minimization. This is the first work that offers a general and rigorous identifiability condition for unknown chain components. We also conduct experiments comparing our algorithm's performance against existing baselines.

Introduction

Probabilistic graphical models offer architectures for modeling and representing uncertainties in decision making. From a computational standpoint, graphical representations enable efficient algorithms for inference, e.g., message passing, loopy belief propagation, and other variational inference methods (Kschischang, Frey, and Loeliger 2001). They have found applications in a wide range of domains, e.g., image processing, natural language processing and computational biology; see (Koller and Friedman 2009; Wainwright and Jordan 2008) and references therein for examples.

A typical application of graphical models is to encode causal information. An influential article from (Pearl 1995) elucidated how *Bayesian networks* can be used to represent causal processes and allow identification of causal effects. The graphical structure of a Bayesian network is a directed acyclic graph (DAG). Each node has a functional dependency on its parents, as determined by the graph. A popular way to substantiate Bayesian networks is as a *linear structural equation model (SEM)* where variables that correspond to nodes in the graph are a linear function of their parents' values plus additive independent noise (often Gaussian) (Bollen 1989; Spirtes et al. 2000). (Hoyer et al. 2008) defined the more general *additive noise model* where each node is an arbitrary function of its parents with an additive independent noise.

While Bayesian networks offer a clear conceptual way to model the causal structure of a system, they are in practice very hard to infer from data, as they require knowledge of how every single variable is generated. In applications involving hundreds of variables (e.g., in computational biology), this requirement is unreasonable, particularly because at the end, we may only be interested in causal effects on a few target variables. Furthermore, in SEMs modeled by Bayesian networks, the noise terms of different variables must be independent whereas in real-world systems, correlations can arise for various reasons (e.g., latent confounders). An interesting middle ground is the notion of *chain graphs* (Lauritzen and Wermuth 1989). Here, the variable set is partitioned into *chain components*, and there is a DAG on these chain components. The variables inside each chain component, however, are connected by undirected edges, not directed ones. See Figure 1 for an illustration. Thus, chain graph models interpolate between directed (causal) models and undirected (probabilistic) models.

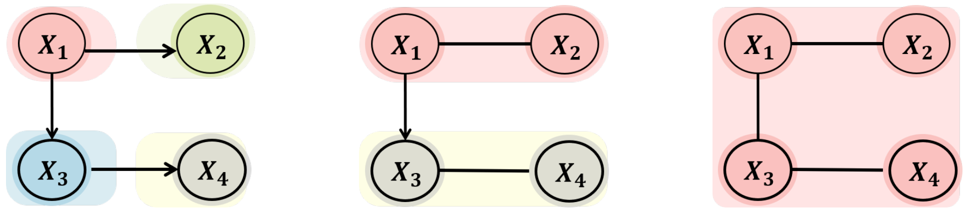
There are several prevalent interpretations of chain graph models, namely the Lauritzen-Wermuth-Frydenberg (LWF) (Lauritzen and Wermuth 1989; Frydenberg 1990), Alternative Markov Property or Andersson-Madigan-Perlman (AMP) (Andersson, Madigan, and Perlman 2001), and Multivariate Regression (MVR) (Cox and Wermuth 1993). They differ in the conditional independence relations implied by the graphical structure. In this work, we restrict ourselves to the linear AMP model, which is very natural from a generative viewpoint. Let $\mathcal{C}$ be an AMP chain graph[*] on n nodes. Suppose the nodes are partitioned into chain components $\{\tau\}$. Then, we say that a random variable $X \in \mathbb{R}^n$ is *generated by* $\mathcal{C}$ if for every chain component τ:

$$X_\tau = M_\tau X_{\mathrm{Pa}(\tau)} + Z_\tau \tag{1.1}$$

where X_τ is X restricted to τ, $\mathrm{Pa}(\tau) = \{v : \exists u \in \tau, v \to_{\mathcal{C}} u\}$, M_τ is a matrix satisfying:

$$(M_\tau)_{uv} \neq 0 \implies v \to_{\mathcal{C}} u,$$

[*] See Notations and Preliminaries section for formal definitions.



(a) $\mathcal{C}_1$ (b) $\mathcal{C}_2$ (c) $\mathcal{C}_3$

Figure 1: Chain graphs. Each shaded region is a maximal chain component.

and Z_τ is independent from $X_{\mathrm{Pa}(\tau)}$ and is a multivariate Gaussian drawn from $N(\mathbf{0}, \Sigma_\tau)$ where Σ_τ satisfies:

$$(\Sigma_\tau^{-1})_{uv} \neq 0 \implies u -_{\mathcal{C}} v$$

The last condition ensures that $N(0, \Sigma_\tau)$ is Markovian with respect to the undirected induced subgraph $\mathcal{C}_\tau$ on τ. One may also consider the additive noise AMP formulation where each

$$X_\tau = f_\tau(X_{\mathrm{Pa}(\tau)}) + Z_\tau, \tag{1.2}$$

the noise Z_τ is as above, and the function f_τ is arbitrary, provided it satisfies the directed graph structure:

$$\frac{(\partial f_\tau)_u}{\partial X_v} \neq 0 \implies v \to_{\mathcal{C}} u.$$

The directed edges of the AMP chain graph form a Bayesian network structure on the chain components, while for each τ, the undirected induced subgraph $\mathcal{C}_\tau$ describes a Gaussian Markov model for $X_\tau \mid X_{\mathrm{Pa}(\tau)}$.

In this work, we focus on the question of *identifiability* of chain graph models. That is, given knowledge of the distribution of X, can we recover the AMP chain graph $\mathcal{C}$ generating X? Moreover, can we recover $\mathcal{C}$ in polynomial time? For Bayesian networks[†], the study of identifiability has received sustained attention for more than two decades. In general, the problem is computationally hard (Chickering 1996), but by making faithfulness or related assumptions, many sets of researchers (e.g., (Spirtes et al. 2000; Chickering 2002; Zhang and Spirtes 2016; Raskutti and Uhler 2018)) have shown that the underlying DAG can be recovered up to its Markov equivalence class. This is quite unsatisfactory as the faithfulness assumption becomes too restrictive in the presence of finite sample error and the DAG is not uniquely identifiable. In a different line of work, (Peters and Bühlmann 2014) showed that $\mathcal{C}$ is exactly identifiable for linear Gaussian SEMs if all the noise terms have equal variance. (Ghoshal and Honorio 2017, 2018) and (Park and Kim 2020) established identifiability conditions for linear SEMs even with unknown heterogeneous error variances. Most recently, (Park 2020) extended these conditions to additive noise models, while (Gao, Ding, and Aragam 2020) further generalized to arbitrary Bayesian networks. See also (Eberhardt 2017) and (Glymour, Zhang, and Spirtes 2019) for other perspectives.

We extend these identifiability conditions from DAGs to linear AMP chain graph models. Our main contributions are:

(i) **Additive noise AMP with known chain component decomposition**: We give a general class of identifiability conditions (generalizing the equal variance condition for linear SEMs) that imply identifiability of the DAG on a known collection of chain components. For instance, the DAG is identifiable if the determinant of the conditional covariance of a chain component τ given τ's parents[‡] is the same for all τ. More generally, it is sufficient for this determinant to be monotonically non-decreasing with respect to a topological order on the chain components. The same is true if the trace or the permanent satisfies the monotonicity condition.

(ii) **AMP with unknown chain component decomposition**: We give an identifiability condition for recovering the chain components as well as the DAG for the standard AMP chain graph model. Informally, the requirement is quite natural: the variables in each chain component should be tightly correlated, while as a whole, each chain component should have large variance conditioned on its parents. More formally, the conditions are that:

  (a) If S is a proper subset of a chain component τ:
  $$\det(\mathrm{Cov}(X_S \mid X_{\tau \setminus S}, X_{\mathrm{Pa}(\tau)})) < 1$$

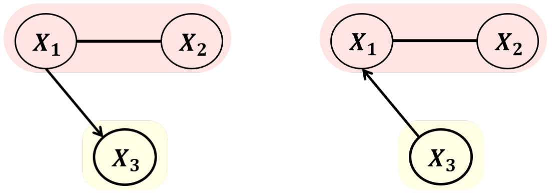
  (b) $\det(\mathrm{Cov}(X_\tau \mid X_{\mathrm{Pa}(\tau)}))$ is greater than 1 and equal for all chain components τ. (Again, similar to (i) above, one can relax “equal” to “monotonically non-decreasing”.)

In our conditions, the determinant of the covariance matrix of Gaussians plays a central role, and this is for good reason. If $X \sim N(0, \Sigma)$ is an n-dimensional Gaussian, then $\det(\Sigma)$ is the *generalized variance* of X and is related to its *differential entropy*. Namely, the differential entropy of X is $\frac{1}{2}(\log\det(\Sigma) + n\log(2\pi e))$; see, e.g., (Krause, Singh, and Guestrin 2008; Yu 2015). So, one can interpret condition (a) above as: If S is a proper subset of τ, its differential entropy conditioned on $\tau \setminus S$ and τ's parents is smaller than a threshold. Similarly, the first part of condition (b) can be restated as: If S equals τ, the differential entropy of S conditioned on its parents is larger than a threshold.

These identifiability conditions come with polynomial time algorithms. Notably, our algorithm for recovering the chain components in (ii) above involves a non-trivial submodular function minimization, in contrast to the more straightforward algorithms known for identifying linear SEMs and Bayesian networks (Park 2020; Gao, Ding, and

[†] For Gaussian graphical models, identifiability reduces to finding the inverse of the covariance matrix.

[‡] Note that if the generating equation is $X_\tau = B \cdot X_{Pa(\tau)} + Z_\tau$, where $Z_\tau = (\mathbf{0}, \Sigma_\tau)$ is the noise, then $\mathrm{Cov}(X_\tau | X_{Pa(\tau)}) = \Sigma_\tau$.



(a) $\mathcal{C}_1$ (b) $\mathcal{C}_2$

Figure 2: Chain graph identifiability: how to determine which of these graphs is generating a given joint distribution $P(X_1, X_2, X_3)$?

Aragam 2020) under analogous conditions. The conditions in (i) and (ii) that determinants of residual covariances are equal is especially relevant where each chain component corresponds to the same physical system (e.g., in time series data).

Technical Overview

In this section, we describe some of the intuition behind our identifiability conditions.

Known chain components. Consider Figure 2 which shows two chain graphs $\mathcal{C}_1$ and $\mathcal{C}_2$; the question is to determine which of these graphs is generating a given joint distribution (X_1, X_2, X_3). In $\mathcal{C}_1$, let $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}(0, \Sigma_1)$, and $X_3 = \beta_1 X_1 + Z$, where $\beta_1 \neq 0$ and $Z \sim \mathcal{N}(0, \sigma^2)$. In $\mathcal{C}_2$, let $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \beta_2 \\ 0 \end{pmatrix} X_3 + Z$ where $\beta_2 \neq 0$, $Z \sim \mathcal{N}(0, \Sigma_2)$ and $X_3 \sim \mathcal{N}(0, \sigma^2)$. Assume $\mathrm{Det}(\Sigma_1) = \mathrm{Det}(\Sigma_2) = \sigma^2$, so that in both models, the determinant of the covariance of each chain component conditioned on its parents is σ^2.

We claim one can distinguish between $\mathcal{C}_1$ and $\mathcal{C}_2$ based on the generated distribution. Our algorithm first finds the chain component τ minimizing $\det(\mathrm{Cov}(X_\tau))$. Note that for $\mathcal{C}_1$, using the independence of Z: $\mathrm{Cov}(X_3) = \beta_1^2 \,\mathrm{Cov}(X_1) + \mathrm{Cov}(Z) \succ \mathrm{Cov}(Z)$, assuming[§] $\mathrm{Cov}(X_1) \succ 0$. Hence, $\det(\mathrm{Cov}(X_3)) > \det(Z) = \sigma^2 = \det(\mathrm{Cov}(X_{12}))$. On the other hand for $\mathcal{C}_2$, $\det(\mathrm{Cov}(X_{12})) > \sigma^2 = \det(\mathrm{Cov}(X_3))$. Thus, the chain component with the smallest determinant of the covariance can be identified as the first in a topological ordering. This can be understood as the uncertainty level of the parents is less than its children. Once the first chain component is known, we can select the second by choosing the one that minimizes the determinant of its covariance conditioned on the first chain component, and so on. It suffices to find the topological order because as described in (Gao, Ding, and Aragam 2020), one can identify the directed edges by standard variable selection methods.

Note that the only property we used of the determinant is that $\det(A + B) > \det(A)$ if B is strictly positive definite. This property holds not only for the determinant but for many natural matrix functions. For example for any i, the diagonal entries $(A + B)_{ii} > A_{ii}$ when A and B are positive definite. Carrying out the same logic as above but now using projection to diagonal entries instead of determinants implies that the chain component DAG is identifiable when all the individual variables have equal variance, extending the result of (Peters and Bühlmann 2014) to chain graphs. In fact, there is a large class of functions called "generalized matrix functions" that satisfy the desired super-additivity condition and hence result in identifiability conditions for the DAG on chain components.

Unknown chain components. Consider again $\mathcal{C}_1$ from Figure 2, but suppose now that we do not have the chain component partitioning. Let (X_1, X_2, X_3) be generated as described above. In addition to imposing the condition that $\det(\Sigma_1) = \sigma^2$, we now also require that: (i) $\det(\mathrm{Cov}(X_1|X_2))$ and $\det(\mathrm{Cov}(X_2 \mid X_1))$ are[¶] strictly less than 1, and (ii) σ^2 is strictly greater than 1.

Now, we can show that

$$\det(\mathrm{Cov}(X_{12})) = \min_{S \subseteq \{1,2,3\}} \det(\mathrm{Cov}(X_S)).$$

Observe that $\det(\mathrm{Cov}(X_3)) > \det(\mathrm{Cov}(X_{12}))$ already follows from the earlier discussion. We now compare $\det(\mathrm{Cov}(X_{12}))$ to $\det(\mathrm{Cov}(X_1))$ and $\det(\mathrm{Cov}(X_2))$. We use the fact that:

$$\det(\mathrm{Cov}(X_{12})) = \det(\mathrm{Cov}(X_1)) \cdot \det(\mathrm{Cov}(X_2 \mid X_1)).$$

This follows from standard facts about multivariate Gaussians. From our assumption $\det(\mathrm{Cov}(X_2 \mid X_1)) < 1$, we get that $\det(\mathrm{Cov}(X_1)) > \det(\mathrm{Cov}(X_{12}))$. The same holds for $\det(\mathrm{Cov}(X_2))$. Finally, we need to show that $\det(\mathrm{Cov}(X_{123})) > \det(\mathrm{Cov}(X_{12}))$. Again, we can invoke the above fact:

$$\det(\mathrm{Cov}(X_{123})) = \det(\mathrm{Cov}(X_{12})) \cdot \det(\mathrm{Cov}(X_3 \mid X_{12})).$$

Our conclusion follows from the assumption $\sigma^2 > 1$.

For a general chain graph, it similarly follows that the non-empty set S minimizing $\det(\mathrm{Cov}(X_S))$ is the topologically smallest. We can identify the next component by conditioning on the components already discovered, which results in a Gaussian on the rest, and then finding a non-empty subset with conditional covariance matrix of smallest determinant. This algorithm can be implemented efficiently. The reason is that for any positive definite $n \times n$-matrix M, the function $F(S) = \log\det(M[S, S])$, where $M[S, S]$ is the submatrix on rows and columns indexed by $S \subseteq [n]$, is *submodular*. F, as noted earlier, corresponds to the differential entropy of a Gaussian vector with covariance M, which is a submodular function, plus an additional modular term. The problem of submodular function minimization has a long and rich history, beginning with the seminal works of (Grötschel, Lovász, and Schrijver 1981, 2012) and continuing to the current day (Iwata, Fleischer, and Fujishige 2001; Schrijver 2000; Lee, Sidford, and Wong 2015; Dadush, Végh, and Zambelli 2018; Jiang 2021). Thus, we can invoke any of these known polynomial-time algorithms for submodular function minimization to recover the chain components in topological order.

[§] In this work, we make the assumption everywhere that all covariance matrices are strictly positive definite.

[¶] $\det(\mathrm{Cov}(X_2 \mid X_1))$ is well defined, since (X_1, X_2) are jointly Gaussian, and hence, for any choice of x_1, $\mathrm{Cov}(X_2 \mid X_1 = x_1)$ is the same.

Related Work

Learning DAG Models. The literature on learning pure DAG models is vast. One popular approach is to exploit the constraints imposed by Markov structure, e.g., the PC algorithm and its variants, like Fast Causal Inference (FCI), Really Fast Causal Inference (RFCI) and Cyclic Causal Discovery (CCD) (Spirtes, Glymour, and Scheines 2000; Spirtes et al. 2000; Richardson 2013; Colombo et al. 2011; Tom Claassen and Smyth 2013; Harris and Drton 2013; Colombo and Maathuis 2014) under different assumptions. Another important class of algorithms aims to maximize a score function over the space of DAG's, such as Greedy Equivalence Search (GES) (Chickering 2002; Ramsey et al. 2017; Nandy et al. 2018) and a recent line of work that formulates score maximization as a continuous optimization problem (e.g., (Zheng et al. 2018, 2020; Wei, Gao, and Yu 2020)). This latest direction has resulted in algorithms that learn the DAG structure with deep learning methods (e.g., (Yu et al. 2019; Lachapelle et al. 2020; Wang et al. 2020)).

DAG Identifiability. A probability distribution may be Markov with respect to many Bayes networks; so for exact identifiability, one needs to impose more structural constraints on the DAG model. For Structural Equation Models (SEM's), identifiability can be established by leveraging asymmetries between variable pairs (Shimizu et al. 2006; Mooij et al. 2016), restricting SEMs to having additive noise, such as linear non-Gaussian acyclic model (LiNGAM) (Shimizu et al. 2006), general additive noise models (Peters et al. 2014), Post-nonlinear model (PNL) (Zhang et al. 2016), or equal and unknown error variance (Peters and Bühlmann 2014; Ghoshal and Honorio 2017; Eberhardt 2017; Ghoshal and Honorio 2018; Chen, Drton, and Wang 2019; Glymour, Zhang, and Spirtes 2019; Park and Kim 2020; Park 2020; Gao, Ding, and Aragam 2020).

Learning AMP Chain Graph Models. AMP chain graphs, our focus in this work, have been less widely studied than pure DAG models and more in the statistics literature than computer science. Informally speaking, (Peña 2015) showed that any AMP model can be viewed as arising from a DAG causal model subject to selection bias. (Levitz, Perlman, and Madigan 2001) introduced a pathwise separation criterion to characterize conditional independence relations in AMP chain graphs. (Roverato 2005; Studenỳ, Roverato, and Štěpánová 2009; Peña 2017a) studied the equivalence classes of chain graph models, and (Peña 2018) provided a factorization for positive distributions that are Markov with respect to an AMP chain graph. (Drton et al. 2009) showed that the AMP conditional independence relations may lead to non-smooth models for discrete variables. (Peña 2014b, 2016) investigated extensions to the AMP model, e.g., the marginal AMP model (MAMP) that is a common generalization of AMP and MVR. When the chain graph structure is known, (Drton and Eichler 2006) proposed an algorithm for maximum likelihood estimation of the model parameters. (Peña 2012, 2014a; Peña and Gomez-Olmedo 2016) proposed PC-LIKE, a constraint based algorithm under faithfulness assumptions for learning the structure of AMP and MAMP models. Peña also designed a score-based algorithm for AMP model structure learning similar to the work on additive noise models (Peña 2017b) and an algorithm based on answer set programming (Peña 2016). Recently, (Javidian, Valtorta, and Jamshidi 2020) solved the problem of efficiently finding minimal separating sets in AMP chain graphs and obtained a new decomposition-based structure learning algorithm called LCD-AMP.

Notations and Preliminaries

Probability. We need the following useful fact about conditional covariance. The proof is a simple generalization of the standard proof for the law of total variance.

Fact 1.1 (Law of Conditional Covariance). *If X, Y, Z are random variables with strictly positive distributions with each component having finite second moment, then:*

$$\mathrm{Cov}_X(X \mid Y) = \mathbb{E}_Z[\mathrm{Cov}_X(X \mid Y, Z) \mid Y] + \mathrm{Cov}_Z(\mathbb{E}_X[X \mid Y, Z] \mid Y).$$

The following result yields a very useful decomposition for covariance of normal distributions.

Fact 1.2. *If $X = (X_A, X_B)$ is distributed jointly as a Gaussian $\mathcal{N}(\mathbf{0}, \Sigma)$, then:*

$$\det(\mathrm{Cov}(X)) = \det(\mathrm{Cov}(X_A)) \cdot \det(\mathrm{Cov}(X_B \mid X_A))$$

where $\mathrm{Cov}(X_B \mid X_A) = \mathrm{Cov}(X_B \mid X_A = x_A)$ is independent of x_A.

Chain Graphs. Following conventions in the field, a variable is denoted by an uppercase letter, e.g., X, and its value is denoted by the corresponding lowercase letter, $z \in Z$, where Z is the state space of X. Graphs in this paper contain both directed ('→') and undirected ('—') edges. Below we will further invoke the most central definitions and notations used in this paper. For a general account, we refer the reader to (Lauritzen 1996) and (Edwards 2012).

A chain graph $\mathcal{C}$ consists of a *vertex set* V and an *edge set* $E \in V \times V$. Two vertices joined by an edge are called *adjacent*. A *path* in $\mathcal{C}$ is a sequence of distinct vertices $\langle v_0, \ldots, v_n \rangle$ such that v_{i-1} and v_i are adjacent for all $1 \leq i \leq n$, and is called a *cycle* if $v_n = v_0$. Moreover, a *semi-directed cycle* exists if $v_1 \rightarrow v_2$ is in $\mathcal{C}$ and $v_i \rightarrow v_{i+1}$, $v_i \longleftrightarrow v_{i+1}$ or $v_i - v_{i+1}$ is in $\mathcal{C}$ for all $1 < i < n$. For any subset S, the set of parents of v is denoted as $\mathrm{Pa}(v) := \{v \in V \setminus S \mid v \rightarrow s \in \mathcal{C}$ for some $s \in S\}$, the set of *children* of v is denoted as $\mathrm{Ch}(v) := \{v \in V \setminus S \mid s \rightarrow v \in \mathcal{C}$ for some $s \in S\}$, the set of $neighbours$ is denoted as $\mathrm{Ne}(v) := \{v \in V \setminus S \mid v - s \in \mathcal{C}$ for some $s \in S\}$. A chain graph with no directed edges is an undirected graph (UG), while a chain graph with no undirected edges is a DAG. For vertices $(u, v) \in E$ but $(v, u) \notin E$, we write $u \rightarrow v$, where vertex u is a *parent* of v. If both $(u, v) \in E$ and $(v, u) \in E$, we denote it by $u - v$, which means u is a *neighbor* of v. The vertex set of a chain graph can be partitioned into *chain components* $\{\tau \mid \tau \in \mathcal{T}\}, (V = \cup_{(\tau \in \mathcal{T})} \tau)$. Edges within chain components are undirected whereas edges between two chain components are directed. A $source$ node is any node X_τ such that $\mathrm{Pa}(X_\tau) = \emptyset$. A $sink$ node is any node X_τ such that $\mathrm{Ch}(X_\tau) = \emptyset$. The chain components τ of a chain graph are the connected components of the undirected graph obtained by removing all directed edges from the chain graph. In a DAG, all chain components are singletons. For $S \subseteq V$, $\mathcal{C}_S$ denotes the induced subgraph on S.

By taking into account the directed connections of chain components, an AMP chain graph admits a topological ordering of its chain components. For statistical identifiability of chain graph $\mathcal{C}$, we will consider it sufficient to learn the partition into chain components $\tau_1, \ldots, \tau_t$, and a topological ordering $\prec$ such that $\tau_j \to \tau_k \implies \tau_j \prec \tau_k$. One can learn the directed and undirected edges using standard parameter estimation algorithms.

Matrix Algebra. Our identifiability condition in the case of known chain components is in terms of positive and super-additive families, which we define next.

Definition 1.3. *Let $\mathbb{C}_n$ denote the cone of $n \times n$ positive semidefinite matrices. We say that a real-valued function $d_n : \mathbb{C}_n \to \mathbb{R}$ is* positive and super-additive *if: (i) $d_n(A) > 0$ for all positive definite matrices A, and (ii) for all positive semidefinite matrices A, B:*

$$d_n(A + B) \geq d_n(A) + d_n(B).$$

A positive and super-additive family *is a collection of functions $f_n : \mathbb{C}_n \to \mathbb{R}$, each of which is positive and super-additive.*

We have several examples of families of positive and super-additive functions:

- Clearly, the projection on any diagonal element and the matrix trace function are positive and super-additive.
- By Minkowski's determinant theorem (see, e.g., (Marcus and Minc 1992)), it known that for all $A, B \in \mathbb{C}_n$: $(\det(A+B))^{1/n} \geq (\det(A))^{1/n} + (\det(B))^{1/n}$. Hence, $\{\det^{1/n} : \mathbb{C}_n \to \mathbb{R}\}$ is positive and super-additive.
- For χ an irreducible character on a subgroup H of S_n (the permutation group on n elements), define the *generalized matrix function* with respect to H and χ as:

  $$d_\chi^H(A) = \sum_{\sigma \in H} \chi(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

  where $A = (a_{i,j})$. (Schur 1918) showed that $d_\chi^H(A) > 0$ for all positive definite A. It is also known (e.g., (Merris 1997), p. 228) that they satisfy the super-additivity condition. Hence, the determinant[‖], permanent, and the Hadamard matrix function (product of diagonal entries) all form positive and super-additive families.

Identifiability with Known Chain Component Decomposition

In this section, we give a general class of conditions which are sufficient to ensure that the DAG structure of the chain graph is identifiable from data generated by it. Here, the chain component decomposition $\mathcal{D}$ is already known to the algorithm. $\mathcal{D}$ consists of t disjoint maximal chain components that partition the variable set.

We formulate our results for general AMP chain graph models. They will immediately imply the conditions for additive noise AMP models mentioned in the Introduction.

Algorithm 1: Our algorithm for learning the topological order of a chain graph with chain component decomposition $\mathcal{D}$ of size t.

```
$\mathcal{A}, P \leftarrow \emptyset$;
$i \leftarrow 0$;
while $|\mathcal{A}| \neq t$ do
    $\tau_i \leftarrow \arg\min_{\tau \in \mathcal{C} \setminus \mathcal{A}} d_{|\tau|}(\mathbb{E}[\mathrm{Cov}(X_\tau \mid X_P)])$;
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\tau_i\}$;
    $P \leftarrow P \cup \tau_i$;
    $i \leftarrow i + 1$;
Return the ordering $(\tau_1, \ldots, \tau_t)$
```

Theorem 1.4. *Suppose the random variable X is generated by an AMP-CG $\mathcal{C}$ with known chain component decomposition $\mathcal{D}$. Then, $\mathcal{C}$ is identifiable from P if there exists a topological ordering π of $\mathcal{C}$ and a positive and super-additive family $\{d_n : \mathbb{C}_n \to \mathbb{R}\}$ such that:*

$$\begin{aligned} d_{|\tau|} \left(\underset{X_{\mathrm{Pa}(\tau)}}{\mathbb{E}} \underset{X_\tau}{\mathrm{Cov}}(X_\tau \mid X_{\mathrm{Pa}(\tau)}) \right) \leq \\ d_{|\tau'|} \left(\underset{X_{\mathrm{Pa}(\tau')}}{\mathbb{E}} \underset{X_{\tau'}}{\mathrm{Cov}}(X_{\tau'} \mid X_{\mathrm{Pa}(\tau')}) \right) \end{aligned} \tag{1.3}$$

for any two chain components τ, τ' where $\tau \prec_\pi \tau'$.

The following corollary is immediate.

Corollary 1.5. *Suppose X corresponds to an additive noise model generated by a chain graph $\mathcal{C}$, i.e.:*

$$X_\tau = f_\tau(X_{\mathrm{Pa}(\tau)}) + Z_\tau,$$

where the noise term Z_τ is independent of $X_{\mathrm{Pa}(\tau)}$, for all chain components τ of $\mathcal{D}$.

Then, given the chain component decomposition, a topological ordering of $\mathcal{D}$ is identifiable from X if there exists a topological ordering π of $\mathcal{D}$ such that

$$\det(\mathrm{Cov}(Z_\tau)) \leq \det(\mathrm{Cov}(Z_{\tau'}))$$

for all chain components $\tau \prec_\pi \tau'$.

Non-parametric algorithm. We give a finite-sample version algorithm using the determinant as d_n. One can estimate $\det(\mathbb{E}[\mathrm{Cov}(X_\tau \mid X_P)])$ by (i) using a non-parametric regressor $\widehat{F}_{\tau,P}(X_P)$ to estimate $\mathbb{E}[X_\tau \mid X_P]$ with n_1 samples, and (ii) using a plug-in estimator on n_2 samples:

$$\det \left(\frac{1}{n_2} \sum_{i=1}^{n_2} \left((X_\tau^{(i)})^{\otimes 2} - \widehat{F}_{\tau,P}^{\otimes 2}(X_P^{(i)}) \right) \right)$$

Using standard non-parametric regularity conditions, we can lower bound the probability of the algorithm recovering the true topological order. Note that the result does not depend on the particular choice of the estimator $\widehat{F}_{\tau,P}$ as long as it is asymptotically consistent. Due to space constraints, a detailed statement is deferred to the appendix.

[‖] The super-additivity of the determinant is also directly implied by the super-additivity of $\det^{1/n}$.

General Identifiability

In this section, we establish identifiability conditions for recovering both the chain components as well as the DAG structure of chain graphs from the generated probability distribution. Here, by identifiability, we mean that the partitioning into chain components and the topological order on the chain components are uniquely specified. The exact set of directed and undirected edges can then be recovered using standard variable selection methods (as described in Appendix A of (Gao, Ding, and Aragam 2020)).

Algorithm 2: Infinite sample algorithm for learning the topological order of a chain graph with unknown chain components.

```
1 $P \leftarrow \emptyset$;
2 $i \leftarrow 1$;
3 $\tau_1 = \arg\min_{S \subseteq V, S \neq \emptyset} \det(\mathrm{Cov}(X_S))$ ;
4 $P \leftarrow P \cup \tau_1$;
5 while $V \setminus P \neq \emptyset$ do
6     $\tau_i \leftarrow \arg\min_{S \subseteq V \setminus P, S \neq \emptyset} \det(\mathrm{Cov}(X_S \mid X_P))$;
7     $P \leftarrow P \cup \tau_i$;
8     $i \leftarrow i + 1$;
9 Return the topological sort $(\tau_1, \ldots, \tau_i)$
```

Theorem 1.6. *Suppose the random variable X is generated by an AMP-CG $\mathcal{C}$ with unknown structure. Then, $\mathcal{C}$ is identifiable from X if the following three conditions hold:*

(i) For all chain components τ and all non-empty proper subsets $S \subset \tau$:

$$\det(\mathrm{Cov}(X_s \mid X_{\tau \setminus s}, X_{\mathrm{Pa}(\tau)})) < 1.$$

(ii) For all chain components τ:

$$\det(\mathrm{Cov}(X_\tau \mid X_{\mathrm{Pa}(\tau)})) > 1.$$

(iii) There is a topological order π on the chain components such that for all $\tau \preceq_\pi \tau'$. :

$$\det(\mathrm{Cov}(X_\tau \mid X_{\mathrm{Pa}(\tau)})) \leq \det(\mathrm{Cov}(X_{\tau'} \mid X_{\mathrm{Pa}(\tau')})).$$

Informally speaking, for any subset S, given its complementary set and parents union of τ in $\mathcal{C}$, we require the variables in each chain component to be tightly correlated. Besides, given the union of the parents of chain components τ, we require the clustered variables in each chain component to have large generalized variance. The third condition is the same one imposed in the identifiability with known chain component decomposition section.

There is a geometric way to view the conditions in Theorem 1.6, which substantiates the intuition that they require each chain component to cluster together while having large variance as a whole. Recall that for any matrix M, $\det(M)$ corresponds to the volume of the parallelepiped spanned by the rows of M. Let the chain components be denoted $\tau_1, \ldots, \tau_k$ in a topological order. For $i = 1, \ldots, k$, let M_i denote the covariance matrix of $X_{\tau_i} \mid X_{\tau_1 \cup \cdots \cup \tau_{i-1}}$, and let M denote the full covariance matrix, $\mathrm{Cov}(X_{\tau_1 \cup \cdots \cup \tau_k})$. From Fact 1.2,

$$\det(M) = \det(M_1) \cdots \det(M_k). \tag{1.4}$$

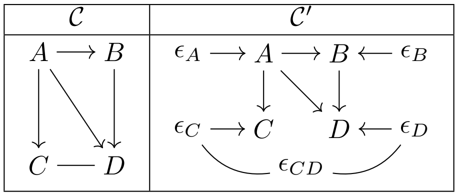


Figure 3: Synthetic data generation. Undirected edges correspond to correlated noise.

Let V_i denote the set of row vectors of M_i, and we identify V_i with the parallelepiped it spans. Due to Equation 1.4, we can view each V_i as residing in a subspace orthogonal to the spans of other V_j's, so that their volumes just multiply with each other. (Alternatively, construct a block diagonal matrix M' where the i'th block on the diagonal is M_i; clearly, $\det(M) = \det(M')$.) In this language, Condition (ii) in Theorem 1.6 says that the volume of each V_i is more than 1, and condition (iii) says that the volumes are non-decreasing with i. Condition (i) says that for any V_i, the volume of any sub-parallelepiped is larger than the volume of the whole. Intuitively, this means that the vectors in V_i form very small angles with each other, so that the volumes keep decreasing as more vectors are added.

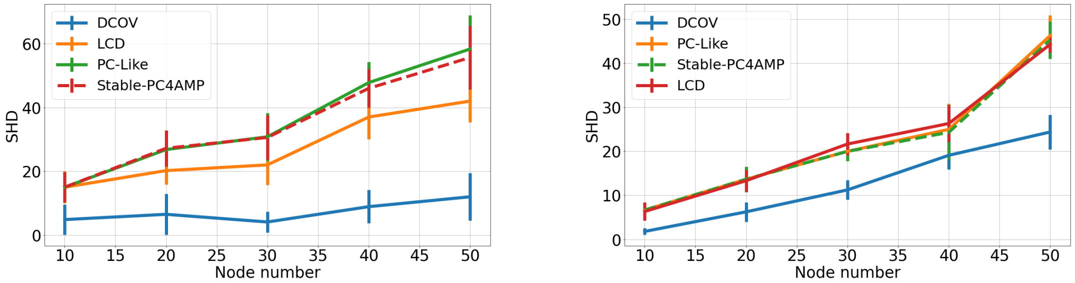
Computational Efficiency. It is known that Algorithm 2 can be implemented in polynomial time. This is because the optimization problems in lines 3 and 5 of the pseudocode correspond to submodular function minimization, as explained earlier. A slight non-triviality is that the optimization is over all non-empty sets instead of over all sets. However, it is well-known how to reduce this to unconstrained minimization (e.g., see Section 4.1 of (Gurjar and Rathi 2020)).

Experiments

In this section, we compare the performance of Algorithm 1 and Algorithm 2 on synthetic datasets to state-of-the-art methods for AMP chain graph structure learning[**]. Recall that as we showed in Theorem 1.4, the DAG on the chain components of an AMP chain graph is identifiable if (1.3) is satisfied for a positive and super-additive family d_τ. Here, we let d_τ be the determinant operator, and hence dub our algorithm as Determinant of Covariance (DCOV).

Synthetic Data Generation. To generate the chain graph $\mathcal{C}$, in our first step, an undirected graph $\mathcal{G}$ with n nodes is generated by using the Erdős Rényi (ER) model with an expected neighbor size $s = 2$ and then symmetrizing. Given the number of chain components c, we split the interval $[1, n]$ into c equal-length sub-intervals $[I_1, \ldots, I_c]$ so that variable sets for each sub-interval forms chain components $\tau_1, \ldots, \tau_c$. Meanwhile, for any (i, j) pair, we set $\mathcal{C}_{i,j} = 0$ if $\exists i \in I_\ell, j \in I_m, \ell > m$. Given the binary adjacency matrix $\mathcal{C}$, we generate the matrix M of edge weights by $M_{i,j} \sim U(-1.5, -0.5] \cup U[0.5, 1.5)$ if $\mathcal{C}_{i,j} \neq 0$ and $M_{i,j} = 0$ otherwise.

[**] Code is available at https://github.com/YohannaWANG/DCOV



(a) Algorithm 1 (known chain components)

(b) Algorithm 2 (unknown chain components)

Figure 4: SHD performance (lower is better)

The observational i.i.d. data $X_\tau = M_\tau X_{\mathrm{Pa}(\tau)} + Z_\tau$ is generated with a sample size $n = 1000$ and a variable size $d \in \{10, 20, 30, 40, 50\}$. Z_τ is an independent multivariate Gaussian drawn from $N(0, \Sigma_\tau)$ where Σ_τ is generated randomly with $\det(\Sigma_\tau) = 1$, satisfying the assumption of Corollary 1.5. Figure 3 illustrates how the synthetic AMP chain graph data is generated.

Baseline Algorithms. We compare our DCOV method against the PC-LIKE (Peña 2012, 2014a; Peña and Gomez-Olmedo 2016), STABLE-PC4AMP (Javidian, Valtorta, and Jamshidi 2020), and LCD algorithm (Learn Chain Graphs via Decomposition) (Ma, Xie, and Geng 2008). We use default parameters among those baseline algorithms in order to avoid skewing the results in favour of any particular algorithm as a result of hyperparameter tuning[††]. All the baseline algorithms above are implemented using R-packages (licensed under GPL-2 or GPL-3) such as `ggm` (Marchetti et al. 2006), `pcalg` (Kalisch et al. 2012), `mgcv` (Wood and Wood 2015), `np` (Racine and Hayfield 2020), and `lcd` (Ma, Xie, and Geng 2008). We use `rpy2` (Gautier 2012) to access R-packages from Python and ensure that all algorithms can be compared in the same environment. The results are averaged over 20 independent repetitions. The experiments were conducted on an Intel Core i7-9750H 2.60GHz CPU.

Implementation of DCOV. We implement Algorithm 2 using the Matlab toolbox `Submodular Function Optimization` (Krause 2010). Each iteration of Algorithm 1 and Algorithm 2 estimates the conditional covariance of the remaining chain components using the finite-sample algorithm mentioned earlier. Like (Gao, Ding, and Aragam 2020), we run a *gam* regression to estimate conditional expectations. We set the p-value with significance level of 0.001 for determining the parents of the node.

Performance Evaluation Metrics. We evaluate the performance of the proposed algorithms in terms of the four measurements, namely, true positive rate (TPR), false positive rate (FPR), accuracy (ACC), and structural hamming distance (SHD) that are commonly used in (Javidian, Valtorta, and Jamshidi 2020; Colombo and Maathuis 2014; Ma, Xie, and Geng 2008).

Agnostic learning When the chain components are unknown, our theoretical results treat the case that the data is realized following the conditions in Theorem 1.6. A straightforward question is: how the algorithm performs when the condition is violated? To answer this question, we conduct agnostic learning experiments by showing the experiment results based on the following conditions:

1. *Chain graph experiments:* Take the opposite condition from Theorem 1.6, where $\det(\mathrm{Cov}(X_\tau \mid X_{\mathrm{Pa}(\tau)})) \leq 1$.
2. *DAG experiments:* Evaluate the performance of our algorithms on synthetic Directed Acyclic Graph (DAG) data;

(a) The variance for each node is equal and > 1;
(b) The variance for each node is uniformly in $[0.5, 1.5]$;

Summary of Experiment Results. As shown in Figure 4, DCOV, under both known and unknown chain component conditions, shows superior performance compared with all other baselines by wide margins. This is because the identifiability condition we proposed provides a correctness guarantee for the recovery of chain-graph structures. Surprisingly, in the DAG structure learning task, if the condition in Theorem 1.6 holds, our unknown chain graph structure learning algorithm (Algorithm 2) can correctly identify the special one-node chain component structures. It also shows superior performance over all other baseline methods. Besides, in our chain graph agnostic learning experiments, when node number increases, although SHD is still lower than other baseline algorithms, the ACC, TPR, and FPR performances are relatively worse. Furthermore, we also conduct agnostic learning experiments on DAG structures. Since our proposed condition does not hold in this case, in the worst condition, Algorithm 2 can wrongly treat all the nodes in a DAG graph as one chain component. This leads to the highest SHD and FPR, and lower ACC performance in our experiment results. We also evaluate the performance of Algorithm 1 on four real Gaussian Bayesian networks from R package `bnlearn` (Scutari 2009). The **ECOLI70** graph provided by (Schafer and Strimmer 2005) contains 46 nodes and 70 edges. The **MAGIC-NIAB** graph from (Scutari et al. 2014) contains 44 nodes and 66 edges. The **MAGIC-IRRI** graph contains 64 nodes and 102 edges. The experimental details are available in the supplementary material. One limitation of this work is the lack of real datasets that can be modeled by chain graphs.

[††] The implementation of baseline algorithms is available at https://github.com/majavid/AMPCGs2019.

Conclusion

In this work, we address the problem of recovering linear AMP chain graph in polynomial time from observational data, and we proposed two algorithms for both known and unknown chain components to handle the problem. In our experiments, we implemented our algorithms over both known and unknown chain components. As future work, we are also interested in exploring a score-based approach for chain graph structure learning from observational data.

References

Andersson, S. A.; Madigan, D.; and Perlman, M. D. 2001. Alternative Markov properties for chain graphs. *Scandinavian journal of statistics*, 28(1): 33–85.

Bollen, K. A. 1989. Measurement models: The relation between latent and observed variables. *Structural equations with latent variables*, 179–225.

Chen, W.; Drton, M.; and Wang, Y. S. 2019. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4): 973–980.

Chickering, D. M. 1996. Learning Bayesian networks is NP-complete. In *Learning from data*, 121–130. Springer.

Chickering, D. M. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov): 507–554.

Colombo, D.; and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1): 3741–3782.

Colombo, D.; Maathuis, M. H.; Kalisch, M.; and Richardson, T. S. 2011. Learning high-dimensional DAGs with latent and selection variables. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 850–850. AUAI Press.

Cox, D. R.; and Wermuth, N. 1993. Linear dependencies represented by chain graphs. *Statistical science*, 204–218.

Dadush, D.; Végh, L. A.; and Zambelli, G. 2018. Geometric rescaling algorithms for submodular function minimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 832–848. SIAM.

Drton, M.; and Eichler, M. 2006. Maximum likelihood estimation in Gaussian chain graph models under the alternative Markov property. *Scandinavian journal of statistics*, 33(2): 247–257.

Drton, M.; et al. 2009. Discrete chain graph models. *Bernoulli*, 15(3): 736–753.

Eberhardt, F. 2017. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3(2): 81–91.

Edwards, D. 2012. *Introduction to graphical modelling*. Springer Science & Business Media.

Frydenberg, M. 1990. The chain graph Markov property. *Scandinavian Journal of Statistics*, 333–353.

Gao, M.; Ding, Y.; and Aragam, B. 2020. A polynomial-time algorithm for learning nonparametric causal graphs. *Advances in Neural Information Processing Systems*, 33.

Gautier, L. 2012. rpy2: A simple and efficient access to R from Python, 2012. *URL http://rpy. sourceforge. net/rpy2. html.*

Ghoshal, A.; and Honorio, J. 2017. Learning identifiable Gaussian Bayesian networks in polynomial time and sample complexity. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6460–6469.

Ghoshal, A.; and Honorio, J. 2018. Learning linear structural equation models in polynomial time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, 1466–1475. PMLR.

Glymour, C.; Zhang, K.; and Spirtes, P. 2019. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10: 524.

Grötschel, M.; Lovász, L.; and Schrijver, A. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2): 169–197.

Grötschel, M.; Lovász, L.; and Schrijver, A. 2012. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media.

Gurjar, R.; and Rathi, R. 2020. Linearly Representable Submodular Functions: An Algebraic Algorithm for Minimization. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Harris, N.; and Drton, M. 2013. PC algorithm for nonparanormal graphical models. *Journal of Machine Learning Research*, 14(11).

Hoyer, P.; Janzing, D.; Mooij, J. M.; Peters, J.; and Schölkopf, B. 2008. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21: 689–696.

Iwata, S.; Fleischer, L.; and Fujishige, S. 2001. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4): 761–777.

Javidian, M. A.; Valtorta, M.; and Jamshidi, P. 2020. AMP Chain Graphs: Minimal Separators and Structure Learning Algorithms. *Journal of Artificial Intelligence Research*, 69: 419–470.

Jiang, H. 2021. Minimizing convex functions with integral minimizers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 976–985. SIAM.

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

Koller, D.; and Friedman, N. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Krause, A. 2010. SFO: A toolbox for submodular function optimization. *Journal of Machine Learning Research*, 11: 1141–1144.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2).

Kschischang, F. R.; Frey, B. J.; and Loeliger, H.-A. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2): 498–519.

Lachapelle, S.; Brouillard, P.; Deleu, T.; and Lacoste-Julien, S. 2020. Gradient-Based Neural DAG Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lauritzen, S. L. 1996. *Graphical models*, volume 17. Clarendon Press.

Lauritzen, S. L.; and Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics*, 31–57.

Lee, Y. T.; Sidford, A.; and Wong, S. C.-w. 2015. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 1049–1065. IEEE.

Levitz, M.; Perlman, M. D.; and Madigan, D. 2001. Separation and completeness properties for AMP chain graph Markov models. *Annals of statistics*, 1751–1784.

Ma, Z.; Xie, X.; and Geng, Z. 2008. Structural learning of chain graphs via decomposition. *Journal of Machine Learning Research*, 9: 2847.

Marchetti, G. M.; et al. 2006. Independencies induced from a graphical Markov model after marginalization and conditioning: the R package ggm. *Journal of Statistical Software*, 15(6): 1–15.

Marcus, M.; and Minc, H. 1992. *A survey of matrix theory and matrix inequalities*, volume 14. Courier Corporation.

Merris, R. 1997. *Multilinear algebra*. Crc Press.

Mooij, J. M.; Peters, J.; Janzing, D.; Zscheischler, J.; and Schölkopf, B. 2016. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1): 1103–1204.

Nandy, P.; Hauser, A.; Maathuis, M. H.; et al. 2018. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A): 3151–3183.

Park, G. 2020. Identifiability of Additive Noise Models Using Conditional Variances. *Journal of Machine Learning Research*, 21(75): 1–34.

Park, G.; and Kim, Y. 2020. Identifiability of Gaussian linear structural equation models with homogeneous and heterogeneous error variances. *Journal of the Korean Statistical Society*, 49(1): 276–292.

Pearl, J. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4): 669–688.

Peña, J. M. 2012. Learning AMP chain graphs under faithfulness. *arXiv preprint arXiv:1204.5357*.

Peña, J. M. 2014a. Learning marginal AMP chain graphs under faithfulness. In *European Workshop on Probabilistic Graphical Models*, 382–395. Springer.

Peña, J. M. 2014b. Marginal AMP chain graphs. *International Journal of Approximate Reasoning*, 55(5): 1185–1206.

Peña, J. M. 2015. Every LWF and AMP chain graph originates from a set of causal models. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 325–334. Springer.

Peña, J. M. 2016. Alternative Markov and Causal Properties for Acyclic Directed Mixed Graphs. In *The 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016), New York City, NY, USA, June 25-29, 2016*.

Peña, J. M. 2017a. Identification of strong edges in AMP chain graphs. *arXiv preprint arXiv:1711.09990*.

Peña, J. M. 2017b. Learning Causal AMP Chain Graphs. In *Advanced Methodologies for Bayesian Networks*, 33–44. PMLR.

Peña, J. M. 2018. Reasoning with alternative acyclic directed mixed graphs. *Behaviormetrika*, 45(2): 389–422.

Peña, J. M.; and Gomez-Olmedo, M. 2016. Learning marginal AMP chain graphs under faithfulness revisited. *International Journal of Approximate Reasoning*, 68: 108–126.

Peters, J.; and Bühlmann, P. 2014. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1): 219–228.

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

Racine, J. S.; and Hayfield, T. 2020. Package 'np'.

Ramsey, J.; Glymour, M.; Sanchez-Romero, R.; and Glymour, C. 2017. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2): 121–129.

Raskutti, G.; and Uhler, C. 2018. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1): e183.

Richardson, T. S. 2013. A discovery algorithm for directed cyclic graphs. *arXiv preprint arXiv:1302.3599*.

Roverato, A. 2005. A unified approach to the characterization of equivalence classes of DAGs, chain graphs with no flags and chain graphs. *Scandinavian Journal of Statistics*, 32(2): 295–312.

Schafer, J.; and Strimmer, K. 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1).

Schrijver, A. 2000. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2): 346–355.

Schur, I. 1918. Über endliche Gruppen und hermitesche Formen. *Mathematische Zeitschrift*, 1(2): 184–207.

Scutari, M. 2009. Learning Bayesian networks with the bnlearn R package. *arXiv preprint arXiv:0908.3817*.

Scutari, M.; Howell, P.; Balding, D. J.; and Mackay, I. 2014. Multiple quantitative trait analysis using Bayesian networks. *Genetics*, 198(1): 129–137.

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

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. Causation, prediction, and search. Adaptive computation and machine learning.

Spirtes, P.; Glymour, C. N.; Scheines, R.; and Heckerman, D. 2000. *Causation, prediction, and search*. MIT press.

Studenỳ, M.; Roverato, A.; and Štěpánová, Š. 2009. Two operations of merging and splitting components in a chain graph. *Kybernetika*, 45(2): 208–248.

Tom Claassen, T. H., Joris M. Mooij; and Smyth, P. 2013. Learning Sparse Causal Models is not NP-hard. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press.

Wainwright, M. J.; and Jordan, M. I. 2008. *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Wang, Y.; Menkovski, V.; Wang, H.; Du, X.; and Pechenizkiy, M. 2020. Causal discovery from incomplete data: a deep learning approach. *arXiv preprint arXiv:2001.05343*.

Wei, D.; Gao, T.; and Yu, Y. 2020. DAGs with No Fears: A Closer Look at Continuous Optimization for Learning Bayesian Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wood, S.; and Wood, M. S. 2015. Package 'mgcv'. *R package version*, 1: 29.

Yu, Y.; Chen, J.; Gao, T.; and Yu, M. 2019. DAG-GNN: DAG Structure Learning with Graph Neural Networks. In *International Conference on Machine Learning*, 7154–7163.

Yu, Y.-L. 2015. Submodular Analysis, Duality and Optimization. http://www.cs.cmu.edu/~yaoliang/mynotes/submodular.pdf. Accessed: 2021-02-18.

Zhang, J.; and Spirtes, P. 2016. The three faces of faithfulness. *Synthese*, 193(4): 1011–1027.

Zhang, K.; Wang, Z.; Zhang, J.; and Schölkopf, B. 2016. On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2): 13.

Zheng, X.; Aragam, B.; Ravikumar, P. K.; and Xing, E. P. 2018. DAGs with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 9472–9483.

Zheng, X.; Dan, C.; Aragam, B.; Ravikumar, P.; and Xing, E. 2020. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 3414–3425. PMLR.