UCLA

UCLA Previously Published Works

Title

Structural Discovery with Partial Ordering Information for Time-Dependent Data with Convergence Guarantees

Permalink

https://escholarship.org/uc/item/2kn0k83m

Journal

Journal of Computational and Graphical Statistics, ahead-of-print(ahead-of-print)

ISSN

1061-8600

Authors

Lin, Jiahe

Lei, Huitian

Michailidis, George

Publication Date

2024

DOI

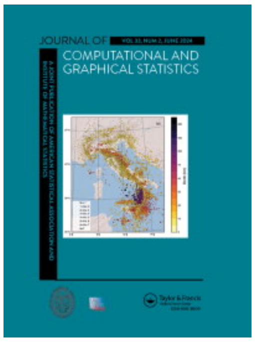
10.1080/10618600.2023.2301097

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library University of California



Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

Structural Discovery with Partial Ordering Information for Time-Dependent Data with Convergence Guarantees

Jiahe Lin, Huitian Lei & George Michailidis

To cite this article: Jiahe Lin, Huitian Lei & George Michailidis (09 Feb 2024): Structural Discovery with Partial Ordering Information for Time-Dependent Data with Convergence Guarantees, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2023.2301097

To link to this article: https://doi.org/10.1080/10618600.2023.2301097

View supplementary material

Published online: 09 Feb 2024.

Submit your article to this journal

Article views: 80

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ucgs20

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2024, VOL. 00, NO. 0, 1–10
https://doi.org/10.1080/10618600.2023.2301097

Taylor & Francis
Taylor & Francis Group

Structural Discovery with Partial Ordering Information for Time-Dependent Data with Convergence Guarantees

Jiahe Lin[a], Huitian Lei[b], and George Michailidis[c]

[a]Machine Learning Research, Morgan Stanley, New York, NY; [b]Lyft, Inc., San Francisco, CA; [c]Department of Statistics & Data Science, University of California, Los Angeles, CA

ABSTRACT
Structural discovery among a set of variables is of interest in both static and dynamic settings. In the presence of lead-lag dependencies in the data, the dynamics of the system can be represented through a structural equation model (SEM) that simultaneously captures the contemporaneous and temporal relationships amongst the variables, with the former encoded through a directed acyclic graph (DAG) for model identification. In many real applications, a partial ordering amongst the nodes of the DAG is available, which makes it either beneficial or imperative to incorporate it as a constraint in the problem formulation. This article develops an algorithm that can seamlessly incorporate a priori partial ordering information for solving a linear SEM (also known as Structural Vector Autoregression) under a high-dimensional setting. The proposed algorithm is provably convergent to a stationary point, and exhibits competitive performance on both synthetic and real datasets. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received May 2023
Accepted November 2023

KEYWORDS
Constrained optimization; Identifiability; Multi-block ADMM; Structural equation models

1. Introduction

Learning the interactions among a set of time series is a topic of interest and pertinent to applications in economics, functional genomics, neuroscience, environmental sciences and social media analysis (see, e.g., recent review papers by Runge et al. 2019; Vowels, Camgoz, and Bowden 2022 and references therein). Dynamic Bayesian Networks (DBN) capture conditional dependence relationships among a set of variables evolving over time and hence constitute a natural modeling framework for this task (Ghahramani 1997). They extend the notion of static graphical models over a set of variables as a function of time, wherein the structural relationships are encoded through a directed acyclic graph (DAG), and the temporal relationships are captured through lag dynamics. The problem of learning the parameters of DBNs from data has received significant attention in the literature; for example, see Scanagatta, Salmerón, and Stella (2019) and references therein. Further, when the relationships and dynamics are assumed to be linear, DBN can be expressed as a *structural* Vector Autoregressive (SVAR) model (see also (1)), that has been studied in the econometrics literature (Lütkepohl 2005; Kilian and Lütkepohl 2017). However, the focus in the latter line of work has been on a small set of variables, whereas new application areas typically involve a large number of time series (i.e., of high dimension). Note that although the linearity assumption may occasionally be somewhat restrictive, linear models remain relevant and appealing in many real-world settings, due to its interpretability and parsimonious representation when used as a working model.

In many applications, selected prior information is available for the structural relationships among variables. For example, in functional genomics there are known transcription factors that act only as regulators of other genes; analogous information is available for certain macroeconomic indicators, corresponding to the fact that as a group they cannot be descendent nodes to some other ones in a DAG. Therefore, it is important to incorporate such prior information in learning algorithms for the purpose of structural discovery.

In this work, we develop an algorithm for estimating the parameters of large-scale SVAR models, which can incorporate the partial ordering information in a seamless way.

1.1. Related Work

We provide a brief review on existing approaches in the literature for learning the parameters of SVAR models. The key issue on the identifiability of its model parameters and the challenges it poses is presented in Section 2.

Estimation of SVAR. Classical work largely lies in the econometrics domain where methods have been developed primarily for fixed dimension SVAR models; for example, Fry and Pagan (2011), Stock and Watson (2016), and Kilian and Lütkepohl (2017) and references therein. These methods ubiquitously start from the reduced VAR representation (see also (2)), then recover the structural parameter by imposing restrictions on the error covariance structure to achieve model identification. Recent developments on the topic amount to considering the

CONTACT George Michailidis gmichail@umich.edu Department of Statistics & Data Science, University of California, Los Angeles, CA.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

© 2024 American Statistical Association and Institute of Mathematical Statistics

structural component that captures variables' contemporaneous inter-dependencies as a DAG and perform "causal" search or estimation. To that end, recent SVAR estimation methods are largely an extension of their respective DAG estimation counterparts, by considering a formulation that additionally incorporates the lag terms. Estimation is done by either jointly considering the structural and the lag components or through a two-stage procedure that relies on residuals from the projection onto the lag space; for example, Hyvärinen et al. (2010), Moneta et al. (2011), Malinsky and Spirtes (2018), and Pamfil et al. (2020). Some of these approaches can be extended to high dimensions.

Estimation of DAGs. In light of the close connection between the SVAR and DAG problems, we briefly review approaches in estimating the latter next. There are three lines of work around this task: the first and most general one—in the absence of any additional prior information—include approaches that leverage greedy search algorithms over the space of the DAGs (Chickering 2002; Tsamardinos, Brown, and Aliferis 2006), those that rely on conditional independence tests (Spirtes et al. 2000; Kalisch and Bühlman 2007) or likelihood-based ones (Van de Geer and Bühlmann 2013; Aragam and Zhou 2015). Note that the computational complexity of estimating a DAG from observational data is superexponential in the number of nodes/variables (Robinson 1977) and thus many of them are greedy in nature and do not scale well even for moderate size problems involving 20–50 variables. More recently, optimization-based approaches (Zheng et al. 2018) and nonlinear ones relying on neural networks have also been considered (Yu et al. 2019; Lachapelle et al. 2019). The second and rather restrictive approach, is that a *total topological ordering* for the nodes in the DAG $\mathcal{G}$ is known either from the literature or extensive experimental work on related settings (see, e.g., discussion in Markowetz (2010) for applications in functional genomics, and Rahman et al. (2023) for an application in agriculture). The problem effectively boils down to estimating whether an edge is present as all potential parent nodes are known (Shojaie and Michailidis 2010). The third and least explored category is to have limited information on the structural relationships among the variables in the form of a *partial ordering* of the underlying nodes in $\mathcal{G}$; such information is available in a variety of applications and two examples are given in Section 5. The notion of partial ordering will be formally defined in Section 2.

In Reisach, Seiler, and Weichwald (2021), the authors report that the performance of DAG estimation using continuous structural learning methods (or equivalently, optimization-based approaches; for example, NOTEARS (Zheng et al. 2018)) can be sensitive to the data scale as measured by the concept of "varsortability" introduced in that paper; specifically, these methods may face issues in the absence of high varsortability (i.e., when the marginal variance of the data is informative of the topological ordering). Given that selected recent methods for time-series data are built upon their DAG estimation counterparts (e.g., Dynotears (Pamfil et al. 2020) as an extension to NOTEARS), such susceptibility permeates. On the other hand, data in real-world applications may not possess strong varsortability, which may render estimates based on such methods unreliable.

Contribution. The main challenges in estimating high-dimensional SVAR models include identification of the model parameters and developing efficient algorithms for large-scale models. To this end, the key contribution of this article is the development of a scalable and provably convergent algorithm to estimate the parameters of a SVAR model in a high dimensional regime. Additionally, the devised algorithm can seamlessly incorporate prior partial ordering information in the optimization problem formulation. Finally, note that despite being an optimization-based approach, the algorithm in this work does not face the same issue and is robust to data normalization. See in-depth discussion in Appendix D.2.

The remainder of the article is organized as follows: Section 2 gives the problem statement and discusses several key issues pertaining to the model in question, namely stability and model parameter identifiability. We present the proposed algorithm and briefly discuss its convergence property in Section 3, and assess its performance on synthetic and real datasets in Sections 4 and 5, respectively.

2. Problem Statement

Consider a system of p variables $X_t := (X_{t,1}, \dots, X_{t,p})^\top$ for which observations over time are collected. The dynamics of $X_t \in \mathbb{R}^p$ are assumed to be in accordance with the following SVAR with lag dynamics:

$$X_t = \mu + AX_t + B_1 X_{t-1} + \cdots + B_d X_{t-d} + \boldsymbol{\epsilon}_t, \tag{1}$$

wherein $A \in \mathbb{R}^{p \times p}$ captures the structural relationships among the p variables, and $B_j\,(j = 1, \dots, d)$ the "lead-lag" ones. It is further assumed that the error process $\boldsymbol{\epsilon}_t$ is independent and identically distributed across time points with mean zero and diagonal covariance matrix $\Sigma_{\boldsymbol{\epsilon}}$. In practical applications, X_t usually has a nonzero mean and hence the SVAR model in (1) would include an intercept term that can be estimated from the data (see also Lütkepohl 2005). Without loss of generality, we assume X_t is mean-zero and omit the intercept term $\mu \in \mathbb{R}^p$ in (1) in the remainder of this article.

Next, we briefly elaborate on the issue of stability of the X_t process and the identifiability of the model parameters, and also discuss how prior information on the structural relationships between the X variables can be accommodated.

Stability of the process. The SVAR model can be equivalently represented through a *reduced* VAR(1) process as follows

$$\mathcal{X}_t = \Phi \mathcal{X}_{t-1} + \nu_t, \tag{2}$$

where $\mathcal{X}_t := [X_t^\top, X_{t-1}^\top, \dots, X_{t-d+1}^\top]^\top \in \mathbb{R}^{dp}$, $\nu_t := [u_t^\top, \mathbf{0}^\top, \dots, \mathbf{0}^\top]^\top$ with $u_t := (\mathrm{I}_p - A)^{-1}\boldsymbol{\epsilon}_t$; Φ is the transition matrix in the companion form, given by

$$\Phi := \begin{bmatrix} (\mathrm{I}_p - A)^{-1}B_1 & \cdots & (\mathrm{I}_p - A)^{-1}B_{d-1} & (\mathrm{I}_p - A)^{-1}B_d \\ \mathrm{I}_p & \cdots & O & O \\ \vdots & \ddots & \vdots & \vdots \\ O & \cdots & \mathrm{I}_p & O \end{bmatrix} \in \mathbb{R}^{dp \times dp}. \tag{3}$$

A reduced VAR process is stable if $\det(\mathrm{I}_{dp} - \Phi z) \neq 0$, for $z \leq 1$ (Lütkepohl 2005). Based on standard results for reduced

VAR processes (Basu and Michailidis 2015), for $\{X_t\}$ to be stable (stationary), a sufficient condition is given by $\varrho(\Phi) < 1$, with $\varrho(\cdot)$ denoting the spectral radius of a square matrix. Note that to obtain valid estimates of model parameters in (2) one requires X_t to be stable (Hamilton 2020); hence, the ensuing discussion on identifiability of model parameters is confined to such processes.

Identifiability of model parameters. The identification of model parameters $(A, B_1, \dots, B_d)$ of the SVAR model is a key issue that has been extensively discussed in the literature. In particular, the difficulty stems from the contemporaneous dependency among the variables, as encoded by A, which requires at least $p(p+1)/2$ restrictions for it to be recovered if one starts from a reduced VAR representation. In this work, we assume that A corresponds to the adjacency matrix of a directed acyclic graph (DAG), which is equivalent to the existence of some permutation(s) π of the rows of A, such that $\pi(A)$ is a lower triangular matrix. Note that a restricted version of this assumption, namely imposing an a priori lower triangular structure to A based on domain knowledge considerations, has been used in the econometrics literature for identification of fixed dimension SVAR models (Stock and Watson 2016; Kilian and Lütkepohl 2017).

On the other hand, in the absence of temporal dependence, a linear SEM of the form $X = AX + \boldsymbol{\epsilon}, X \in \mathbb{R}^p$ where A encodes the underlying DAG $\mathcal{G}_A$, is not necessarily identifiable. The joint distribution $\mathbb{P}(X)$ of the observed variables is fully determined through the product distribution of the error variable $\mathbb{P}(\boldsymbol{\epsilon})$ and $\mathcal{G}_A$; conversely, however, the DAGs that give rise to the same $\mathbb{P}(X)$ are not unique (Spirtes et al. 2000). In other words, multiple $\mathcal{G}_A$'s can be compatible with $\mathbb{P}(X)$ and thus the parameter A is not uniquely identifiable from observational data without additional assumptions. For the purpose of identifiability, the following assumptions on the error distribution $\mathbb{P}(\epsilon)$ have been considered in the literature: (a) the distribution is non-Gaussian (Shimizu et al. 2006); (b) the distribution is Gaussian with equal variance across its coordinates (Peters and Bühlmann 2014); and (c) the distribution is Gaussian with unequal variances that are weakly monotonically increasing in the true ordering π implied by the DAG $\mathcal{G}_A$ (Park 2020).

In summary, in this work, the identification scheme adopted for the parameters of the SVAR model in (1) encompasses the following assumptions: (a) A is the adjacency matrix of a DAG, and (b) any of the above-mentioned three conditions on the distribution of the error $\mathbb{P}(\epsilon_t)$ hold.

Prior information and partial ordering. As mentioned in Section 1, in this work, the incorporation of prior information into the estimation procedure is enabled, with the former in the form of partial ordering. Formally, consider a (time-invariant) partition of the nodes $X_{t,1}, \dots, X_{t,p}$ into disjoint sets $\mathcal{V}_1, \dots, \mathcal{V}_Q$, with $\mathcal{V}_1 \prec \mathcal{V}_2 \prec \cdots \prec \mathcal{V}_Q$, with $\prec$ denoting a precedence relationship, that is, there cannot be an edge $X_{t,i} \to X_{t,j}$ for $i \in \mathcal{V}_q, j \in \mathcal{V}_{q'}, q' < q$. However, the intra-dependency or ordering of the variables within a set $\mathcal{V}_q, \forall\, q$ is not known and needs to be inferred from the data. In the extreme case where no prior information is available, the partition becomes trivial and all nodes effectively fall into one set.

3. A Provably Convergent Estimation Procedure

For ease of exposition, in this section, we consider the special case where $d = 1$ and let $B \equiv B_1$; the case where $d > 1$ can be readily derived by stacking the lags and transition matrices which then gives the lag-1 representation (see representation in (2) and (3)).

To obtain estimates for model parameters A and B, let $\mathbf{X}_n \in \mathbb{R}^{n\times p}$ denote the sample matrix with observations $\{x_1, \dots, x_n\}$ stack in the rows of $\mathbf{X}_n$; $\mathbf{X}_{n-1}$ is analogously defined. The loss function is $\ell(A, B; \mathbf{X}_n, \mathbf{X}_{n-1}) := \frac{1}{2n}\|\mathbf{X}_n - \mathbf{X}_n A^\top - \mathbf{X}_{n-1}B^\top\|_{\mathrm{F}}^2$, and the optimization problem based on the ℓ_2 loss is formulated as

$$(\widehat{A}, \widehat{B}) := \operatorname*{argmin}_{A,B} \Big\{ \ell(A, B; \mathbf{X}_n, \mathbf{X}_{n-1}) + \mu_A\|A\|_1 + \mu_B\|B\|_1 \Big\}, \quad \text{subject to } A \text{ being acyclic}, \tag{4}$$

with the additional ℓ_1-norm regularization terms inducing sparsity. In the presence of a *partial ordering* on A (prior information), the search space of A can be represented as

$$\mathcal{P}_A := \Big\{ A \in \mathbb{R}^{p\times p} : A_{ij} = 0 \text{ for } (i,j) \in \mathcal{I} \times \mathcal{J} \Big\}, \qquad \text{where} \quad \mathcal{I} \times \mathcal{J} \subseteq \{1, \dots, p\} \times \{1, \dots, p\};$$

$A_{ij} = 0 \Leftrightarrow j \notin \mathrm{pa}(i)$, that is, node j cannot be a parent of node i in the DAG representation. In the extreme case, $\mathcal{I} \times \mathcal{J}$ can be a null set, corresponding to the case where no prior information is available. As it can be seen later, our proposed algorithm can readily consume such partial ordering information and perform estimation in the restricted subspace $\mathcal{P}_A \subseteq \mathbb{R}^{p\times p}$.

3.1. The Proposed Algorithm

To solve (4), we leverage the results in Yuan et al. (2019), where acyclicity can be enforced through polyhedral constraints and the formulation can be solved via difference-convex (DC) programming and the augmented Lagrangian method of multipliers (ADMM). Concretely, Theorem 1 in Yuan et al. (2019) states that A is acyclic, if and only if the following $p^3 - p^2$ constraints are satisfied for some $\boldsymbol{\lambda} = [\lambda_{ij}] \in \mathbb{R}^{p\times p}$:

$$\lambda_{ik} + \mathbb{I}(j \neq k) - \lambda_{jk} \geq \mathbb{I}(A_{ij} \neq 0); \quad i, j, k = 1, \dots, p, i \neq j. \tag{5}$$

Together with the partial ordering information, by considering the truncated ℓ_1-function $J_\tau(z) := \min(\frac{|z|}{\tau}, 1)$, $\tau \to 0$ as a surrogate for the indicator function, and introducing $\boldsymbol{\xi} = [\xi_{ijk}] \in \mathbb{R}^{p\times p\times p}, \xi_{ijk} \geq 0$ that translate inequality constraints to equality ones, the optimization problem can be written as

$$\begin{aligned} &\min_{A,B,\lambda} \Big\{ \ell(A, B; \mathbf{X}_n, \mathbf{X}_{n-1}) + \mu_A\|A\|_1 + \mu_B\|B\|_1 \Big\}, \\ &\text{subject to } \lambda_{ik} + \mathbb{I}(j \neq k) - \lambda_{jk} = J_\tau(A_{ij}) + \xi_{ijk}, \\ &\qquad A \in \mathcal{P}_A, \quad \xi_{ijk} \geq 0; \quad i, j, k = 1, \dots, p, i \neq j. \end{aligned} \tag{6}$$

Note that in the case where $A_{ij} = 0$ is a priori enforced, the corresponding constraint in (5) can be simplified to $\lambda_{ik} + \mathbb{I}(j \neq k) - \lambda_{jk} = \xi_{ijk}$, which effectively is a relaxation to the right hand side for (i,j). This is conceptually compatible with the nature

Algorithm 1: Solving for (6) via alternate update between $(A, B, \boldsymbol{\lambda})$ and $\boldsymbol{w}$.

```
Input: Data matrices $\mathbf{X}_n, \mathbf{X}_{n-1}$; search space $\mathcal{P}_A$; and hyperparameters $\mu_A$, $\mu_B$ and $\tau$
while not converging do
    • $\boldsymbol{w}$-update: update $\boldsymbol{w}$ according to $w_{ij} \leftarrow \mathbb{I}(|A_{ij}| < \tau)$;
    • $(A, B, \boldsymbol{\lambda})$-update: update $(A, B, \boldsymbol{\lambda})$ by solving for (7) with a fixed $\boldsymbol{w}$ via ADMM outlined in Algorithm 2.
end
Output: Estimated $\widehat{A}$ and $\widehat{B}$.
```

$$\begin{aligned} \min\nolimits_{A,B,\lambda} \quad & \{\ell(A, B; \mathbf{X}_n, \mathbf{X}_{n-1}) + \mu_A \|A\|_1 + \mu_B \|B\|_1\}, \\ \text{subject to} \quad & \tau\lambda_{ik} + \tau\mathbb{I}(j \neq k) - \tau\lambda_{jk} = |A_{ij}|w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk}, \\ & A \in \mathcal{P}_A, \quad \xi_{ijk} \geq 0; \quad i, j, k = 1, \ldots, p, i \neq j. \end{aligned} \tag{7}$$

of the acyclic constraint, that with $A_{ij} = 0$, the graph could potentially accommodate other edges to be nonzero while still remains acyclic. Similar to Yuan et al. (2019), the formulation in (6) can be solved iteratively leveraging the decomposition $J_\tau(z) = |z|/\tau - \max\{|z|/\tau - 1, 0\}$ and the aid of an indicator matrix $\boldsymbol{w} \in \mathbb{R}^{p\times p}$, whose coordinates are given by $w_{ij} := \mathbb{I}(|A_{ij}| < \tau)$, as outlined in Algorithm 1.

Next, we briefly outline the steps for the $(A, B, \boldsymbol{\lambda})$-update. To handle the non-differentiable parts of the objective function in (7), we introduce $\widetilde{A}$ and $\widetilde{B}$, and the augmented Lagrangian function can be written as follows for some scaled variable $\rho > 0$:

$$\begin{aligned} & L_\rho(A, \widetilde{A}, B, \widetilde{B}, \boldsymbol{\lambda}, \boldsymbol{\xi}; U_A, U_B, \boldsymbol{y}) \\ & \quad := \ell(A, B; \mathbf{X}_n, \mathbf{X}_{n-1}) + \mu_A\|\widetilde{A}\|_1 + \mu_B\|\widetilde{B}\|_1 \\ & \quad + \frac{\rho}{2}\|A - \widetilde{A}\|_{\mathrm{F}}^2 + \rho\langle A - \widetilde{A}, U_A\rangle + \frac{\rho}{2}\|B - \widetilde{B}\|_{\mathrm{F}}^2 \\ & \quad + \rho\langle B - \widetilde{B}, U_B\rangle + \mathcal{T}_1 + \mathcal{T}_2, \end{aligned} \tag{8}$$

with

$$\begin{aligned} \mathcal{T}_1 := \frac{\rho}{2}\sum\nolimits_k \sum\nolimits_{i\neq j} \Big(&|\widetilde{A}_{ij}|w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk} - \tau\lambda_{ik} \\ & - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk}\Big)^2, \\ \mathcal{T}_2 := \rho\sum\nolimits_k \sum\nolimits_{i\neq j} y_{ijk}\Big(&|\widetilde{A}_{ij}|w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk} - \tau\lambda_{ik} \\ & - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk}\Big); \end{aligned}$$

$U_A, U_B, \boldsymbol{y}$ are dual variable matrices/tensors. One proceeds with primal descent on $(A, \widetilde{A}, B, \widetilde{B}, \boldsymbol{\lambda}, \boldsymbol{\xi})$ and dual ascent on $(U_A, U_B, \boldsymbol{y})$ as outlined in Algorithm 2; see Appendix A.1 for the exact update of each step. It is worth noting that given the specific form of the augmented Lagrangian, all primal updates possess closed-form minimizers, which empirically aids in fast and stable convergence of the cyclic block-updates.

To conclude this section, we briefly comment on how the partial ordering information is incorporated through block updates. First, note that by introducing $\widetilde{A}$ that separates the non-differentiable part of A, after some algebra, the update of A (while holding $\widetilde{A}, U_A, B$ fixed) can be written as

$$\begin{aligned} A \leftarrow \underset{A\in\mathcal{P}_A}{\arg\min}\, \mathrm{trace}\Big\{&\frac{1}{2}A\big[(\frac{1}{n}\mathbf{X}_n^\top\mathbf{X}_n) + \rho\mathrm{I}_p\big]A^\top \\ & - \big[(\frac{1}{n}\mathbf{V}_n^\top\mathbf{X}_n) + \rho(\widetilde{A} - U_A)\big]A^\top\Big\}, \end{aligned}$$

where $\mathbf{V}_n := \mathbf{X}_n - \mathbf{X}_{n-1}B^\top$. The update is separable for each row of A; additionally, the prior partial ordering information—in the form of restricting the skeleton indices of each row of A to a subset of $\{1, \ldots, p\}$—becomes equivalent to considering only the corresponding column sub-space of the design matrix.

3.2. Convergence Analysis

We provide a brief discussion on the convergence property of the proposed algorithm, while deferring all lemmas and their proofs to Appendix B.

Note that the $\boldsymbol{w}$-update step is straightforward, which effectively boils down to obtaining the complement of the support of the estimated A at each iteration. The ensuing analysis establishes convergence properties of Algorithm 2.

Denote by $\Theta := (A, \widetilde{A}, B, \widetilde{B}, \boldsymbol{\lambda}, \boldsymbol{\xi})$ and $\Psi := (U_A, U_B, \boldsymbol{y})$ the collection of primal and dual variables, respectively.

Proposition 1. Consider a sequence of iterates $(\Theta^{(s)}, \Psi^{(s)})$ generated by Algorithm 2, indexed by s. Then, the sequence converges to a stationary point of the augmented Lagrangian function (8) for any initial point $(\Theta^{(0)}, \Psi^{(0)})$.

We provide some insights on the critical steps required to achieve convergence. The first is a "sufficient descent property"; namely, one needs to find a positive constant η so that two successive iterates of the primal and dual variables satisfy $\eta\|(\Theta^{(s+1)}, \Psi^{(s+1)}) - (\Theta^{(s)}, \Psi^{(s)})\|_{\mathrm{F}}^2 \leq L_\rho(\Theta^{(s)}, \Psi^{(s)}) - L_\rho(\Theta^{(s+1)}, \Psi^{(s+1)})$, $s = 0, 1, \ldots$. This is established in Lemma 3. The second is a subgradient lower bound for the gap between successive iterates; namely, there exists another positive constant γ such that any element $C^{(s)}$ in the subdifferential of $L_\rho(\Theta^{(s)}, \Psi^{(s)})$ satisfies $\|C^{(s+1)}\|_{\mathrm{F}}^2 \leq \gamma\|(\Theta^{s+1)}, \Psi^{(s+1)}) - (\Theta^{(s)}, \Psi^{(s)})\|_{\mathrm{F}}^2$. This is established in Lemma 5 with the aid of Lemma 4. Note that these two requirements are satisfied by most "good" descent algorithms. Further, when the above two properties hold, then the accumulation points of *any* algorithm is a non-empty, compact and connected set (see Remark 5 in Bolte, Sabach, and Teboulle 2014). However, the existence of η, γ *depends on the structure of the specific algorithm used*; the results in Lemmas 2 and 5 show how to obtain them given the structure and updates of the developed multi-block ADMM in Algorithm 2. The last requirement to establish global-convergence-to-a-critical-point of $L_\rho(\cdot)$ does not depend on the structure of

Algorithm 2: Update $(A, B, \boldsymbol{\lambda})$ via multi-block ADMM: a schematic outline

```
Input: Data matrices $\mathbf{X}_n$, $\mathbf{X}_{n-1}$; search space $\mathcal{P}_A$; fixed $\boldsymbol{w}$; hyperparameters $\mu_A$, $\mu_B$, $\tau$ and $\rho$
while not converging do
    (Primal descent)
    • cyclic update on blocks $A, \widetilde{A}, B, \widetilde{B}, \boldsymbol{\lambda}, \boldsymbol{\xi}$ by minimizing (8) w.r.t. the block of interest;
    (Dual ascent)
    • $U_A$-update: $U_A^{(s)} \leftarrow U_A^{(s-1)} + (A^{(s)} - \widetilde{A}^{(s)})$
    • $U_B$-update: $U_B^{(s)} \leftarrow U_B^{(s-1)} + (B^{(s)} - \widetilde{B}^{(s)})$
    • $\boldsymbol{y}$-update: $y_{ijk}^{(s)} \leftarrow y_{ijk}^{(s-1)} + (|\widetilde{A}_{ij}^{(s)}|w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk}^{(s)} - \tau\lambda_{ik}^{(s)} - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk}^{(s)})$
end
Output: Solution $\widehat{A}$ and $\widehat{B}$ to (8)
```

the algorithm used, but on the type of function $L_\rho(\cdot)$. To that end, Lemma 1 shows that the augmented Lagrangian satisfies the Kurdyka-Łojasiewicz property (Kurdyka 1998), which aids in establishing that the sequence of iterates $(\Theta^{(s)}, \Psi^{(s)})$ generated by Algorithm 2 is a Cauchy sequence.

Proof of Proposition 1. Based on the results of Lemmas 4 and 5, we have that $(\Theta^{(s)}, \Psi^{(s)})$ is a bounded sequence and the set of limit points of $(\Theta^{(s)}, \Psi^{(s)})$ when initialized at $(\Theta^{(0)}, \Psi^{(0)})$ is non-empty, respectively. Further, existing results in the literature—in particular, Lemma 5 and Remark 5 in Bolte, Sabach, and Teboulle (2014)—ensure the compactness of the set of limit points of the sequence $(\Theta^{(s)}, \Psi^{(s)})$, when the latter is initialized at $(\Theta^{(0)}, \Psi^{(0)})$. The remainder of the proof follows along the lines of Theorem 1 in Bolte, Sabach, and Teboulle (2014) by using the Kurdyka-Łojasiewicz property of the augmented Lagrangian function, as established in Lemma 1. □

Remark 1. The class of functions that satisfy the Kurduka-Łojasiewicz property is remarkably large and includes many loss functions, regularization terms, as well as polyhedral constraints used in machine learning tasks. Further, the proof strategy is applicable to many algorithms. In this article, we provide the details for a multi-block ADMM algorithm for the non-convex, non-smooth problem arising from the SVAR problem formulation under consideration. Note that such algorithms are used in many other machine learning problems sharing similar features and hence the proof is of general interest. Finally, note that the proposed algorithm exhibits global convergence to a critical point, that is, such convergence is independent of the algorithm's initialization, which is an attractive feature in practical applications.

Empirically, the proposed algorithm exhibits stable convergence; the alternating update between $\boldsymbol{w}$ and $(A, B, \boldsymbol{\lambda})$ usually converges within 10 iterations; the $(A, B, \boldsymbol{\lambda})$-update step that relies on ADMM typically converges within 100 iterations, although during the very first round of the outer update, it often requires more.

Remark 2. Yuan et al. (2019) provide a brief proof for the ADMM-based algorithm developed for reconstructing DAG from iid data. Specifically, the proof assumes that the augmented Lagrangian function is *strongly convex* and appeals to a result in Boyd et al. (2011) to establish convergence of the algorithm. However, note that the augmented Lagrangian function is not strongly convex; additionally, the result in Boyd et al. (2011) only holds for a two-block ADMM algorithm, rather than the multi-block updates used in Yuan et al. (2019) and the current work. Indeed, establishing convergence for multi-block ADMM even for convex functions was challenging and remained open for awhile, as attested in Chen et al. (2016). In this work, the proof of Proposition 1 takes a different route and leverages a road map outlined in Bolte, Sabach, and Teboulle (2014) that establishes the convergence of "descent-type algorithms" and only requires the Kurdyka-Łojasiewicz property of the augmented Lagrangian function, which is significantly weaker.

4. Synthetic Data Experiments

We evaluate the performance of the proposed algorithm and the effectiveness of incorporating the partial ordering information as priors in the estimation through a series of synthetic data experiments.

Settings. The data are generated according to an SVAR model with $d = 2$ lags, that is,

$$X_t = AX_t + B_1X_{t-1} + B_2X_{t-2} + \boldsymbol{\epsilon}_t, \qquad (9)$$
$$\text{where } \mathbb{E}(\boldsymbol{\epsilon}_t) = 0,\ \Sigma_{\boldsymbol{\epsilon}} := \text{cov}(\boldsymbol{\epsilon}_t) = \text{diag}\big(\sigma_1^1, \ldots, \sigma_p^2\big);$$

coordinates of the noise component are independent and potentially heteroscedastic, depending on the distribution from which it is drawn. We consider cases where the system consists of 100 variables, with the structural parameter A exhibiting varying degree of sparsity and the noise component $\boldsymbol{\epsilon}_t$ drawn from different distributions; see Table 1. [1]

Note that to ensure the stability of the process, the spectral radius ϱ of the companion matrix for the corresponding reduced

[1] Recall that as discussed in Section 2, different sets of assumptions have been provided in the literature to guarantee the identifiability of the underlying DAG in the SVAR model. In our experiment setup, we consider settings where the error distribution is either Gaussian with unequal variances that are weakly monotonically increasing (Park 2020), or non-Gaussian Shimizu et al. (2006); as such, they respectively satisfy assumptions (3) and (1) in the aforementioned discussion.

Table 1. Parameter setup for synthetic data experiments.

setting id	p	s_A	(l_A, u_A)	s_{B_1}	s_{B_2}	(l_B, u_B)	σ_i	noise dist
S1	100	0.05	(0.25, 0.9)	0.05	0.02	(1, 3)	Unif[0.8, 2]	Gaussian
S2	100	0.10	(0.25, 0.7)	0.05	0.02	(1, 3)	Unif[0.8, 2]	Gaussian
S3	100	0.05	(0.25, 0.9)	0.05	0.02	(1, 3)	1	Laplace
S4	100	0.10	(0.25, 0.7)	0.05	0.02	(1, 3)	1	Laplace
S5	100	0.05	(0.25, 0.9)	0.05	0.02	(1, 3)	1	Student's-t (df=4)
S6	100	0.10	(0.25, 0.7)	0.05	0.02	(1, 3)	1	Student's-t (df=4)

NOTE: $s_.$ denotes the sparsity level of the corresponding parameter; $(l_., u_.)$ corresponds to the lower and upper bounds of the (initial) draws of the signals; σ_i corresponds to the standard deviation of the coordinates of the noise component.

VAR, that is,

$$\Phi(A, B_1, B_2) := \begin{bmatrix} (\mathrm{I}_p - A)^{-1}B_1 & (\mathrm{I}_p - A)^{-1}B_2 \\ \mathrm{I}_p & O \end{bmatrix}$$

needs to be strictly less than 1 (see also Section 2); to achieve this, we proceed as in the following steps:

1. For transition matrices B_1 and B_2, their skeletons are determined by independent draws from Bernoulli(s_{B_1}) and Bernoulli(s_{B_2}), respectively; nonzero entries are first drawn from $\pm\mathrm{Unif}(l_B, u_B)$, then scaled such that $\varrho\big(\Phi(O, B_1, B_2)\big) = 0.5$[2], where $\Phi(O, B_1, B_2)$ corresponds to the companion matrix of the reduced VAR if one ignores the structural component.
2. For the structural parameter A, to obtain its skeleton subject to the acyclic constraint, each entry in the lower diagonal is drawn independently from Bernoulli(s_A); nonzero entries are then drawn from $\pm\mathrm{Unif}(l_A, u_A)$.
3. Repeat Steps 1 and 2 if $\varrho\big(\Phi(A, B_1, B_2)\big) < 1$ is not satisfied.

In practice, the above procedure gives a set of parameters that yield $\varrho(\Phi(A, B_1, B_2)) \approx 0.95$ within a few trials. A smaller spectral radius can be attained, if one further reduces the signal strength. Once model parameters are generated, we generate $\{X_t\}$ according to (9); the $\boldsymbol{\epsilon}_t$'s are either Gaussian (S1, S2) or Laplace distributed (S3, S4): in the former case, the σ_i's for each coordinate $i = 1, \ldots, p$ are drawn from Unif(0.8, 2) then sorted according to the topological ordering of the nodes as dictated by A; in the latter case, $\sigma_i \equiv 1$. In settings S5 and S6, we additionally consider the case where the noise are generated from t-distribution, to test the robustness of the proposed method in the presence of heavy tails.

For all settings, we run the proposed algorithm on data with sample sizes $n = 50, 100, 200$ and a varying level of available prior information provided through a partial ordering constraint, that is, 10%, 20%, 50% of the complement of the support set, that is, $\{(i,j) : A_{ij} = 0; (i,j) \in \{1,\ldots,p\} \times \{1,\ldots,p\}\}$. Note that the case with $n = 50$ is a rather challenging setting: considering the number of parameters to be estimated, the estimation is "under-powered".

Remark 3. We briefly comment on the sparsity level adopted in the experiment settings. Consider a limiting case where a total topological ordering of the nodes is known a priori; the DAG learning problem reduces to selecting the parent node set by using sparse regression techniques (Reisach, Seiler, and Weichwald 2021). For iid data, under high dimensional scaling, the sparsity level allowed for consistent estimation of the skeleton is $s \sim o\big(\frac{n}{\log(p^2)}\big)$. Further, in the SVAR setting where the data exhibit temporal dependence, the sparsity level is impacted by an additional κ factor that quantifies the temporal dependence, that is: $s \sim o\big(\frac{n}{\kappa^2 \log(p^2)}\big)$, where $\kappa = \frac{M}{m} > 1$ with M and m denoting the maximum and minimum eigenvalues of the spectral density of the time series data under consideration, respectively (see, e.g., Basu and Michailidis 2015, for sub-Gaussian errors). Based on the above, the sparsity considered in the above settings is fairly high, even if a total topological ordering were given. In our experiments, at most some partial topological ordering information is available; consequently, the permissible level of sparsity further reduces when compared to the limiting case.

Performance evaluation. We focus assessment on the structural component A, and specifically on skeleton recovery across different model settings, sample sizes and percentage of prior information provided, as shown in Table 2. Results on the lag-components B and the overall goodness of fit of the algorithm are provided in Appendix D.3. In particular, to understand the impact of incorporating prior information, we report the True Positive Rate (TP, or recall, equivalently) and True Negative Rate (TN, or 1 − false positive rate, equivalently) for support recovery over different sample sizes and prior setups. $\{00, 10, 20, 50\}$ correspond to varying level of partial ordering information—from no prior (00) to 50% (50) of the non-support—given as a prior constraint. Based on the results in Table 2, the main findings are 3-fold: (a) despite similar setups for all other model parameters, model performance is superior in the case where the noise distribution is Laplace/Student's-t compared to Gaussian with monotonically increasing variances, provided that the estimation is moderately powered (e.g., $n = 100, 200$), as manifested by a higher detection of the true skeleton (i.e., true positive rate) without compromising the true negatives; (b) Although the prior partial ordering information is in the form of zero-constraints, it imposes restrictions on the search space of the skeleton and is integrated throughout the estimation process; therefore, the benefit of incorporating prior information is not limited to ruling out false positives, but it can also promote discovery. (c) As one would expect, the estimation becomes more challenging as the graph becomes more dense, as manifested by significantly lower true positive rate, especially for low sample size settings ($n = 50, 100$). Finally, note that the methodology is robust to the presence of heavy tails, as manifested by the overall comparable performance across settings with different noise distributions, provided all else held identical.

[2] Here we set 0.5 as the target spectral radius; however, it typically cannot be attained exactly except for VAR(1).

Table 2. Evaluation for $\widehat{A}$ obtained using **our proposed method**: Results are based on the median of 10 replicates, with the standard deviation of the corresponding metric reported in parentheses.

		00		10		20		50	
	n	TP	TN	TP	TN	TP	TN	TP	TN
S1	50	0.69(0.040)	0.79(0.006)	0.69(0.051)	0.81(0.006)	0.65(0.042)	0.82(0.005)	0.80(0.030)	0.85(0.006)
	100	0.78(0.020)	0.85(0.004)	0.78(0.020)	0.87(0.004)	0.77(0.030)	0.88(0.005)	0.86(0.025)	0.91(0.006)
	200	0.88(0.018)	0.88(0.005)	0.87(0.017)	0.89(0.006)	0.87(0.017)	0.90(0.006)	0.95(0.014)	0.93(0.006)
S2	50	0.56(0.056)	0.72(0.004)	0.53(0.059)	0.74(0.003)	0.56(0.035)	0.76(0.003)	0.73(0.031)	0.77(0.009)
	100	0.73(0.046)	0.74(0.011)	0.71(0.031)	0.76(0.007)	0.73(0.030)	0.78(0.013)	0.84(0.026)	0.81(0.009)
	200	0.84(0.014)	0.83(0.004)	0.83(0.015)	0.84(0.005)	0.84(0.015)	0.86(0.003)	0.93(0.011)	0.89(0.006)
S3	50	0.69(0.035)	0.84(0.007)	0.69(0.041)	0.86(0.006)	0.70(0.026)	0.87(0.005)	0.83(0.031)	0.89(0.005)
	100	0.84(0.015)	0.86(0.005)	0.84(0.017)	0.88(0.005)	0.83(0.020)	0.89(0.004)	0.93(0.020)	0.91(0.005)
	200	0.91(0.011)	0.91(0.002)	0.91(0.012)	0.92(0.002)	0.92(0.012)	0.92(0.003)	0.98(0.008)	0.95(0.004)
S4	50	0.61(0.067)	0.79(0.012)	0.53(0.043)	0.80(0.008)	0.52(0.031)	0.81(0.005)	0.73(0.041)	0.84(0.005)
	100	0.77(0.044)	0.78(0.014)	0.77(0.036)	0.79(0.009)	0.77(0.031)	0.81(0.007)	0.88(0.032)	0.83(0.009)
	200	0.85(0.014)	0.84(0.006)	0.86(0.009)	0.85(0.004)	0.87(0.010)	0.86(0.005)	0.95(0.010)	0.90(0.007)
S5	50	0.68(0.034)	0.86(0.004)	0.69(0.029)	0.87(0.005)	0.69(0.033)	0.88(0.005)	0.76(0.033)	0.90(0.006)
	100	0.85(0.044)	0.84(0.008)	0.86(0.044)	0.85(0.006)	0.86(0.035)	0.86(0.004)	0.91(0.021)	0.89(0.006)
	200	0.89(0.012)	0.86(0.007)	0.89(0.012)	0.87(0.005)	0.89(0.017)	0.88(0.005)	0.97(0.010)	0.91(0.003)
S6	50	0.61(0.042)	0.76(0.005)	0.60(0.045)	0.78(0.004)	0.63(0.033)	0.80(0.005)	0.74(0.055)	0.81(0.005)
	100	0.79(0.067)	0.77(0.010)	0.79(0.037)	0.78(0.005)	0.80(0.058)	0.80(0.009)	0.81(0.042)	0.81(0.008)
	200	0.85(0.013)	0.84(0.005)	0.85(0.015)	0.85(0.004)	0.86(0.014)	0.86(0.004)	0.95(0.006)	0.90(0.004)

Table 3. Evaluation for $\widehat{A}$ obtained using **SVAR-GFCI**.

		00		10		20		50	
	n	TP	TN	TP	TN	TP	TN	TP	TN
S1	50	0.21(0.03)	0.99(0.001)	0.22(0.03)	0.99(0.001)	0.22(0.03)	1.00(0.001)	0.30(0.03)	1.00(0.001)
	100	0.28(0.03)	0.99(0.001)	0.31(0.03)	0.99(0.001)	0.33(0.03)	1.00(0.001)	0.45(0.03)	1.00(0.001)
	200	0.32(0.05)	0.99(0.001)	0.35(0.04)	0.99(0.001)	0.37(0.04)	1.00(0.001)	0.57(0.03)	1.00(0.001)
S2	50	0.06(0.01)	0.99(0.001)	0.07(0.01)	0.99(0.001)	0.08(0.01)	0.99(0.001)	0.11(0.03)	0.99(0.001)
	100	0.09(0.01)	0.99(0.001)	0.10(0.01)	0.99(0.001)	0.11(0.01)	0.99(0.001)	0.16(0.03)	0.99(0.001)
	200	0.09(0.01)	0.99(0.001)	0.10(0.01)	0.99(0.001)	0.11(0.01)	0.99(0.001)	0.16(0.02)	0.99(0.001)
S3	50	0.20(0.03)	0.99(0.001)	0.21(0.03)	0.99(0.001)	0.22(0.03)	1.00(0.001)	0.30(0.03)	1.00(0.001)
	100	0.28(0.03)	0.99(0.001)	0.30(0.03)	0.99(0.001)	0.31(0.03)	0.99(0.001)	0.47(0.03)	1.00(0.001)
	200	0.35(0.05)	0.99(0.001)	0.37(0.04)	0.99(0.001)	0.41(0.04)	1.00(0.001)	0.58(0.03)	1.00(0.001)
S4	50	0.06(0.01)	0.99(0.001)	0.07(0.01)	0.99(0.001)	0.08(0.03)	0.99(0.001)	0.12(0.03)	0.99(0.001)
	100	0.08(0.01)	0.99(0.001)	0.10(0.01)	0.99(0.001)	0.12(0.01)	0.99(0.001)	0.15(0.03)	0.99(0.001)
	200	0.07(0.01)	0.99(0.001)	0.08(0.01)	0.99(0.001)	0.10(0.01)	0.99(0.001)	0.16(0.02)	0.99(0.001)

NOTE: Results are based on the median of 10 replicates, with the standard deviation of the corresponding metric reported in parentheses.

The performance of our proposed algorithm is also benchmarked against SVAR-GFCI (Malinsky and Spirtes 2018) for settings S1–S4, with the latter being a score-based method that uses greedy optimization on the model score to learn the graph, followed by carrying out statistical tests for conditional independence to orient the edges; see Table 3. In particular, we leverage the python implementation of TETRAD,[3]wherein the available prior information can be passed in as an argument.

A major issue with SVAR-GFCI observed under the settings in consideration is its low discovery rate; this may be due to high dimensionality and low sample size, and it is more pronounced for denser graphs. The partial ordering information aids in improved discovery of the graph skeletons, as what one would expect. Further, contrary to the estimates obtained using our proposed method, here we do not observe discrepancy in terms of recovery performance between the two cases, where the noise distribution is Gaussian versus being Laplace.

Comparison with several other methods (e.g., Pamfil et al. 2020; Hyvärinen et al. 2010) whose existing implementation does not readily consume prior information is deferred to Appendix D, where the comparison is only conducted for the case without partial ordering. Additionally, we also include a discussion related to varsortability (Reisach, Seiler, and Weichwald 2021) and additional results to display the impact from data normalization in Appendix D.2.

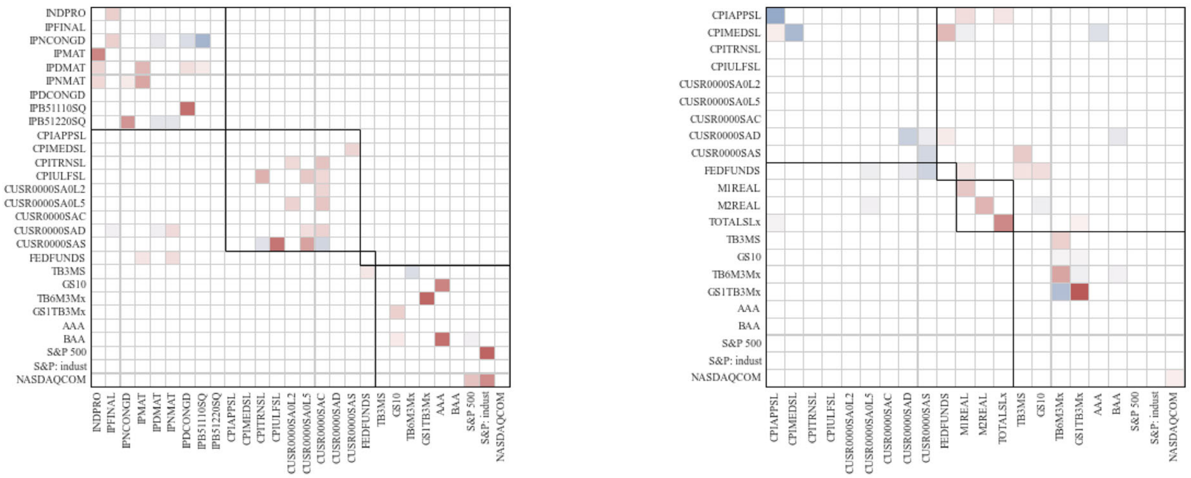
5. Real Data Analysis

To evaluate how our proposed algorithm would perform in real world settings, we consider two applications and examine the structural and temporal components estimated from the proposed method.

5.1. U.S. Macroeconomic Dta

SVAR models are widely used to address various problems in macroeconomic analysis, including the effect of monetary interventions by central banks to the economy (Christiano, Eichenbaum, and Evans 2005). However, small VAR models regularly used in such analyses lead to empirical results that may be contradictory to economic theory tenets (Sims 1980). It has been suggested that large scale SVAR models could overcome such difficulties, however, their identification is typically challenging. The proposed approach offers a principled strategy to use large SVARs.

[3] *https://github.com/cmu-phil/py-tetrad*



(a) A partial view of the estimated $\widehat{A}$ (structural parameter) encompassing industrial production and consumer price indices (tier 1), FFR (tier 2) and market variables (tier 3).

(b) A partial view of the estimated $\widehat{B}_1$ (lag parameter) encompassing consumer price indices, FFR, money stock and market variables.

Figure 1. Partial views of the estimated structural parameter A and lag parameter B_1. Parent variables are depicted in the columns and their descendants are in the rows.

The dataset comprises of a number of US macroeconomic indicators measured at quarterly frequency, sourced from the FRED-QD database (McCracken and Ng 2020). We consider 78 variables spanning the period from 1Q1973 to 2Q2022, totaling 200 observations. These variables encompass several different categories and capture different facets of the economy, with the major ones being industrial production, producer and consumer price indices and their components (tier 1, low tier), Federal Funds Rate (FFR) as an approximation of monetary policy (tier 2, intermediate tier), and several market variables including treasury yields, S&P500 and NASDAQ composite indices (tier 3, high tier); see Bernanke, Boivin, and Eliasz (2005) and discussion therein for inclusion of variables in the model. A prior partial ordering is constructed based on variables' tiers; in particular, we do not allow variables from a higher tier to be the parent nodes of those in a lower tier, that is, the following directional relationship is *prohibited*: {(tier 2, tier 3) → tier 1; tier 3 → tier 1}. This is predicated on the premise that tier 1 variables are "slow moving" and hence not impacted within the same time period by the FFR (tier 2) or "fast moving" variables in tier 3, the latter being sensitive to contemporaneous economic information and shocks (Bernanke, Boivin, and Eliasz 2005). Finally, to ensure stationarity of the time series, we apply the benchmark transformation suggested in McCracken and Ng (2020), which follows from Stock and Watson (2012a, 2012b); further, these time series are de-meaned before they are fed into the model.

We run the proposed algorithm on this dataset, with the hyperparameters μ_A, μ_B selected so that the one-step-ahead predictive RMSE is minimized; recall, that the SVAR model can be expressed in a reduced form as in (2) which gives the recursive relationship for prediction. We set the number of lags $d = 2$, trying to strike a balance between parsimony by not over-expanding the model parameter space and capturing delayed effects through adequate inclusion of lags. The obtained results show that with $d = 2$, the magnitude of the estimated parameters in B_2 is getting significantly smaller than those in B_1.

The heatmap in Figure 1(A) shows the impact of industrial production and consumer prices indices (tier) to the FFR (tier 2) and market variables (tier 3). It can be seen that the FFR is impacted by selected production indices that act as rough proxies of broader economic activity. Further, there are interactions within blocks of related variables, for example, the aggregate industrial production and consumer price indices and their respective constituents. Of particular interest, is the influence exerted by FFR and those variables that influence it with a lag, as seen from Figure 1(B). In particular, FFR impacts consumer price indices positively, which is in accordance with economic theory, which demonstrates its ability in overcoming difficulties in interpretation noted in the literature when small SVAR models were used (see discussion in Sims 1980; Bernanke, Boivin, and Eliasz 2005). Further, it is closely related to the real money stock and treasury yields for both the short and the long tenors, a result in accordance with past analysis (Bańbura, Giannone, and Reichlin 2010).

5.2. DREAM4 Gene Expression Data

Next, we briefly discuss how the proposed algorithm can aid in the task of identifying functional relationships (network inference) between genes from limited size gene expression data. This is a fundamental problem in functional genomics and a comprehensive solution to it requires a large set of expensive "perturbation" (knock-out or knock down) experiments (see discussion in Markowetz (2010)). The DREAM 4 competition provided datasets to test algorithms for such network inference tasks (Marbach et al. 2009; Greenfield et al. 2010), We run the proposed algorithm on a collection of five datasets corresponding to different network topologies from selected organisms, each containing time series (21 time points) for 100 genes with 10 perturbations each. Further, to quantify how gains in performance could be achieved in the presence of partial ordering information (which can be obtained from the literature and experimental work in functional genomics applications), we run the proposed algorithm with and without partial ordering. The partial ordering information is constructed by considering the following three disjoint sets: "regulator", "target", or "free" that the genes are partitioned into. Specifically, based on a "gold standard" network of gene functional relationships, that is taken as the underlying truth and denoted by A with $A_{ij} \neq 0$ corresponding to an edge $j \to i$, let

$$\begin{aligned} \text{regulator} &:= \big\{i : \forall j,\ A_{ij} = 0 \text{ and } \exists j,\ A_{ji} \neq 0\big\}, \\ \text{target} &:= \big\{i : \exists j,\ A_{ij} \neq 0 \text{ and } \forall j,\ A_{ji} = 0\big\}; \end{aligned}$$

that is, the regulator set consists of genes that emit, but have no incoming edges, and the target set contains those that receive, but have no outgoing edges; all other genes including those simultaneously emitting and receiving or neither emitting nor receiving constitute the "free" set. The partial ordering information is constructed such that we prohibit genes in the regulator set to receive, and those in the target set to emit, while imposing no restrictions on those in the free set. In other words, the partial ordering enforces the regulator and the target sets to form a bipartite graph.

We run the model on each of the five network topologies datasets. Specifically, for each run, we set $d = 1$ and select the hyper-parameters (μ_A, μ_B) based on the following procedure: we first run the algorithm over a grid and select the pair (μ_A^*, μ_B^*) that gives the smallest predicted RMSE; then we set $\mu_B \equiv \mu_B^*$ and run the algorithm over a sequence of μ_A's to obtain the ROC/precision-recall curve. Similar to Lu et al. (2021), AUROC and AUPRC are used as performance measures and obtained for each run, with and without the partial ordering information.

Performance of the proposed algorithm without partial ordering information is in the same ballpark range to other linear methods tested in Lu et al. (2021), with AUPRC between 0.10 and 0.20 and AUROC between 0.60 and 0.65; for an extensive analysis see tables in Lu et al. (2021) and follow-up discussion. Note that many methods exhibiting better performance are nonlinear and can accommodate more complex temporal dynamics of gene expression data. In the presence of partial ordering, we notice a 0.10 increase in AUPRC; the gain in AUROC is of similar magnitude, thus, showing the benefits of the proposed method to consume seamlessly such prior information.

Finally, we note that given the availability of a "gold standard" for the DREAM4 datasets under consideration, the calculated varsortability averages around 0.40. This suggests that for real world applications, the data scale can be rather uninformative about the underlying topological ordering of the variables, thus, posing challenges for selected continuous structural learning based methods.[4] This reiterates the need for developing methods for structural discovery in time-series data that are robust to the data scale.

6. Conclusion

The article presents an efficient algorithm to estimate the parameters of a Structural VAR model, in the presence of a priori information that provides partial ordering information for the variables under consideration. The formulated optimization problem is built upon an existing method that estimates a DAG and augments the objective function with the necessary lag terms that encode the temporal dependency. The acyclicity constraints is enforced through a polynomial number of constraints, which can also seamlessly incorporate the partial ordering information. The proposed algorithm is provably convergent to a stationary point. Numerical experiments on synthetic data illustrate the overall competitive performance of the proposed algorithm to a competing method and also the role of the prior information on the accuracy of the results. Finally, applications to macroeconomic and genomic data demonstrate the usefulness of the algorithm in practical settings.

Supplementary Materials

The supplement contains (i) implementation details of the algorithm, (ii) all technical proofs on the convergence of the algorithm, (iii) additional details on the numerical experiments and (iv) a description of the variables used in Section 5.1.

Acknowledgments

The authors thank the editors and two anonymous reviewers for their constructive comments and suggestions.

Authors Contributions

JL proposed and implemented the method, conducted the experiments, compiled the results and drafted the manuscript. HL conducted the experiments partially and proofread the manuscript. GM conceived the project, contributed the proofs, drafted the manuscript and provided the computing resources.

Data Availability Statement

The code repository for this work is available at *https://github.com/GeorgeMichailidis/high-dim-svar-partial-ordering*

Disclosure Statement

The authors declare that they have no financial or nonfinancial interests that relate to the research described in this article.

Funding

GM was partially supported by NSF grant DMS-2348640.

References

Aragam, B., and Zhou, Q. (2015), "Concave Penalized Estimation of Sparse Gaussian Bayesian Networks," *Journal of Machine Learning Research*, 16, 2273–2328. [2]

[4] Note that we have effectively ignored temporal dependency while calculating varsortability, and hence the model could potentially be mildly misspecified. On the other hand, calculation based on simulated data indicates that even though such mis-specification may introduce a minor downward bias to the truth (i.e., the calculated varsortability based on the misspecified model may underestimate the truth), it will not drastically change the varsortability to a large extent and therefore the conclusion still stands.

Bańbura, M., Giannone, D., and Reichlin, L. (2010), "Large Bayesian Vector Auto Regressions," *Journal of Applied Econometrics*, 25, 71–92. [8]

Basu, S., and Michailidis, G. (2015), "Regularized Estimation in Sparse High-Dimensional Time Series Models," *The Annals of Statistics*, 43, 1535–1567. [3,6]

Bernanke, B. S., Boivin, J., and Eliasz, P. (2005), "Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach," *The Quarterly Journal of Economics*, 120, 387–422. [8]

Bolte, J., Sabach, S., and Teboulle, M. (2014), "Proximal Alternating Linearized Minimization for Nonconvex and Nonsmooth Problems," *Mathematical Programming*, 146, 459–494. [4,5]

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011), "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, 3, 1–122. [5]

Chen, C., He, B., Ye, Y., and Yuan, X. (2016), "The Direct Extension of ADMM for Multi-Block Convex Minimization Problems is not Necessarily Convergent," *Mathematical Programming*, 155, 57–79. [5]

Chickering, D. M. (2002), "Learning Equivalence Classes of Bayesian-Network Structures," *Journal of Machine Learning Research*, 2, 445–498. [2]

Christiano, L. J., Eichenbaum, M., and Evans, C. L. (2005), "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy," *Journal of Political Economy*, 113, 1–45. [7]

Fry, R., and Pagan, A. (2011), "Sign Restrictions in Structural Vector Autoregressions: A Critical Review," *Journal of Economic Literature*, 49, 938–960. [1]

Ghahramani, Z. (1997), "Learning Dynamic Bayesian Networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pp. 168–197. [1]

Greenfield, A., Madar, A., Ostrer, H., and Bonneau, R. (2010), "Dream4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models," *PLoS One*, 5, e13397. [8]

Hamilton, J. D. (2020), *Time Series Analysis*, Princeton: Princeton University Press. [3]

Hyvärinen, A., Zhang, K., Shimizu, S., and Hoyer, P. O. (2010), "Estimation of a Structural Vector Autoregression Model Using Non-Gaussianity," *Journal of Machine Learning Research*, 11, 1709–1731. [2,7]

Kalisch, M., and Bühlman, P. (2007), "Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm," *Journal of Machine Learning Research*, 8, 613–636. [2]

Kilian, L., and Lütkepohl, H. (2017), *Structural Vector Autoregressive Analysis*, Cambridge: Cambridge University Press. [1,3]

Kurdyka, K. (1998), "On Gradients of Functions Definable in o-minimal Structures," *Annales de l'institut Fourier*, 48, 769–783. [5]

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. (2019), "Gradient-based Neural DAG Learning," in *International Conference on Learning Representations*. [2]

Lu, J., B. Dumitrascu, I. C. McDowell, B. Jo, A. Barrera, L. K. Hong, S. M. Leichter, T. E. Reddy, and B. E. Engelhardt (2021), "Causal Network Inference from Gene Transcriptional Time-Series Response to Glucocorticoids," *PLoS Computational Biology*, 17, e1008223. [9]

Lütkepohl, H. (2005), *New Introduction to Multiple Time Series Analysis*, Berlin: Springer. [1,2]

Malinsky, D., and Spirtes, P. (2018), "Causal Structure Learning from Multivariate Time Series in Settings with Unmeasured Confounding," in *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery*, pp. 23–47, PMLR. [2,7]

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009), "Generating Realistic in Silico Gene Networks for Performance Assessment of Reverse Engineering Methods," *Journal of Computational Biology*, 16, 229–239. [8]

Markowetz, F. (2010), "How to Understand the Cell by Breaking It: Network Analysis of Gene Perturbation Screens," *PLoS Computational Biology*, 6, e1000655. [2,8]

McCracken, M., and Ng, S. (2020), "FRED-QD: A Quarterly Database for Macroeconomic Research," Technical Report, National Bureau of Economic Research. [8]

Moneta, A., Chlaß, N., Entner, D., and Hoyer, P. (2011), "Causal Search in Structural Vector Autoregressive Models," in *NIPS Mini-Symposium on Causality in Time Series*, pp. 95–114, PMLR. [2]

Pamfil, R., Sriwattanaworachai, N., Desai, S., Pilgerstorfer, P., Georgatzis, K., Beaumont, P., and Aragam, B. (2020), "Dynotears: Structure Learning from Time-Series Data," in *International Conference on Artificial Intelligence and Statistics*, pp. 1595–1605, PMLR. [2,7]

Park, G. (2020), "Identifiability of Additive Noise Models Using Conditional Variances," *Journal of Machine Learning Research*, 21, 1–34. [3,5]

Peters, J., and Bühlmann, P. (2014), "Identifiability of Gaussian Structural Equation Models with Equal Error Variances," *Biometrika*, 101, 219–228. [3]

Rahman, S., Khare, K., Michailidis, G., Martínez, C., and Carulla, J. (2023), "Estimation of Gaussian Directed Acyclic Graphs Using Partial Ordering Information with Applications to Dream3 Networks and Dairy Cattle Data," *The Annals of Applied Statistics*, 17, 929–960. [2]

Reisach, A., Seiler, C., and Weichwald, S. (2021), "Beware of the Simulated DAG! Causal Discovery Benchmarks May be Easy to Game," in *Advances in Neural Information Processing Systems* (Vol. 34), pp. 27772–27784. [2,6,7]

Robinson, R. W. (1977), "Counting Unlabeled Acyclic Digraphs," in *Combinatorial Mathematics V*, ed. C. H. C. Little, pp. 28–43, Berlin: Springer. [2]

Runge, J., Bathiany, S., Bollt, E., Camps-Valls, G., Coumou, D., Deyle, E., Glymour, C., Kretschmer, M., Mahecha, M. D., Muñoz-Marí, J., et al. (2019), "Inferring Causation from Time Series in Earth System Sciences," *Nature Communications*, 10, 2553. [1]

Scanagatta, M., Salmerón, A., and Stella, F. (2019), "A Survey on Bayesian Network Structure Learning from Data," *Progress in Artificial Intelligence*, 8, 425–439. [1]

Shimizu, S., Hoyer, P. O., Hyvärinen, A., Kerminen, A., and Jordan, M. (2006), "A Linear Non-Gaussian Acyclic Model for Causal Discovery," *Journal of Machine Learning Research*, 7, 2003–2030. [3,5]

Shojaie, A., and Michailidis, G. (2010), "Penalized Likelihood Methods for Estimation of Sparse High-Dimensional Directed Acyclic Graphs," *Biometrika*, 97, 519–538. [2]

Sims, C. A. (1980), "Macroeconomics and Reality," *Econometrica: Journal of the Econometric Society*, 48, 1–48. [7,8]

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. (2000), *Causation, Prediction, and Search*, Cambridge, MA: MIT press. [2,3]

Stock, J. H., and Watson, M. W. (2012a), "Disentangling the Channels of the 2007-2009 Recession," Technical Report, National Bureau of Economic Research. [8]

——— (2012b), "Generalized Shrinkage Methods for Forecasting Using Many Predictors," *Journal of Business & Economic Statistics*, 30, 481–493. [8]

——— (2016), "Dynamic Factor Models, Factor-Augmented Vector Autoregressions, and Structural Vector Autoregressions in Macroeconomics," in *Handbook of Macroeconomics* (Vol. 2), pp. 415–525, Saint Louis: Elsevier. [1,3]

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006), "The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm," *Machine Learning*, 65, 31–78. [2]

Van de Geer, S., and Bühlmann, P. (2013), "ℓ_0-penalized Maximum Likelihood for Sparse Directed Acyclic Graphs," *The Annals of Statistics*, 41, 536–567. [2]

Vowels, M. J., Camgoz, N. C., and Bowden, R. (2022), "D'ya like DAGs? A Survey on Structure Learning and Causal Discovery," *ACM Computing Surveys*, 55, 1–36. [1]

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*, pp. 7154–7163, PMLR. [2]

Yuan, Y., Shen, X., Pan, W., and Wang, Z. (2019), "Constrained Likelihood for Reconstructing a Directed Acyclic Gaussian Graph," *Biometrika*, 106, 109–125. [3,4,5]

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* (Vol. 31). [2]