Topology Identification of Directed Graphs via Joint Diagonalization of Correlation Matrices

Yanning Shen, Xiao Fu, Georgios B. Giannakis, and Nicholas D. Sidiropoulos

***Abstract*—Discovering connectivity patterns of directed networks is a crucial step to understand complex systems such as brain-, social-, and financial networks. Several existing network topology inference approaches rely on structural equation models (SEMs). These presume that exogenous inputs are available, which may be unrealistic in certain applications. Recently, an alternative line of work reformulated SEM-based topology identification as a three-way tensor decomposition task. This way, knowing the exogenous input correlation statistics (rather than the exogenous inputs themselves) suffices for network topology identification. The downside is that this approach is computationally expensive. In addition, it is hard to incorporate prior information of the network structure (e.g., sparsity and local smoothness) into this framework, while such prior information may help enhance performance when handling real-world noisy data. The present work puts forth a *joint diagonalizaition* (JD)-based approach to directed network topology inference. JD can be viewed as a variant of tensor decomposition, but features more efficient algorithms, and can readily account for the network structure. Different from existing alternatives, novel identifiability guarantees are derived regardless of the exogenous inputs or their statistics. Three JD algorithms tailored for network topology inference are developed, and their performance is showcased using simulated and real data tests.**

***Index Terms*—Structural equation models, tensor-based model, joint diagonalization, directed network topology inference.**

I. Introduction

Network analytics play an essential role in modeling and understanding the behavior of complex systems, such as financial markets, brains, and genomics, to name just a few. One of the most important tasks in this context is network topology identification [15, Ch. 7]. Prominent among the popular topology inference methods are structural equation models (SEMs) that are capable of capturing causal (directed) dependencies among complex system components [14]. These directional effects are seldom revealed by approaches relying on symmetric associations among endogeneous nodal variables, such as those represented by covariances or correlations; see e.g., [11], [12]. SEMs have well-documented merits in several learning tasks requiring knowledge of the underlying graph topology; see e.g., [3], [8], [13], [24]. In a nutshell, SEMs capture the relationship among observed nodal processes and the unknown causal connectivity. They can also account for exogenous or confounding inputs in observed nodal processes, which turn out to be critical in identifying directional dependencies—a critical task in network analytics [5].

Work in this paper was supported by the National Science Foundation under projects NSF 1711471, 1500713, ECCS 1608961, ECCS 1808159, IIS 1704074, the Army Research office under project ARO W911NF-19-1-0247, and the National Institutes of Health under project NIH 1R01GM104975-01.

One challenge facing SEM identification is that available methods require knowledge of the exogenous inputs that are "injected" per node to stimulate responses. In certain scenarios, measuring exogenous inputs can be costly or impractical, rendering SEM identification infeasible. Take financial networks as an example, where stocks of different companies are nodal measurements and their cross-correlations are edge weights. Publicly traded stock prices (endogenous) are known to depend on investors' purchases of stocks (exogenous inputs), whose details are often unknown to the public for privacy reasons. To cope with the unavailable exogenous inputs, novel tensor-based approaches have been developed in [28], [29] that only require *second-order statistics* of the exogenous inputs. However, there are still several important challenges for this method: i) Even the statistics of exogenous inputs may not be available in some real world applications, e.g., brain networks; ii) prior information of the network structure such as sparsity or smoothness is not readily leveraged by tensor decompositions – but exploiting such prior information is important for combating noise and modeling errors present in real-world data; and iii) the method in [28] employs the canonical polyadic decomposition (CPD), which is not easy to scale up in the context of network inference.

Topology identification has also been studied in graph signal processing; see e.g., [10], [21], [26], where the focus is on identifying undirected network topologies. SEM-based topologies of directed graphs have been also investigated, and found identifiable using observations on all nodes [5]. This result is elegant and plausible, but when nodal measurements are not available (e.g., when there are missing observations or when only some derived information such as the statistics of the measurements is available), it is unclear whether such methods can still work.

The present paper approaches graph topology identification using joint matrix diagonalization, which is a classic tool originally developed for source separation [6], [20], [22], [30]. It will be seen that a topology can be identified by jointly diagonalizing the slabs of a three-way tensor. This tensor is constructed using *second-order statistics* of the nodal measurements, which makes the method naturally robust to missing or noisy nodal measurements. It will be also shown that the statistics of exogenous inputs are no longer needed to guarantee identifiability of the topology—if some other conditions are satisfied. For example, if a small number of anchor nodes whose connectivity patterns (specifically in-links) are known *a priori* are available, then identifiability is ensured by JD solvers. The anchor node assumption can be easily satisfied in certain applications—e.g., in brain networks, where each node represents a certain region of interest. Established

domain knowledge can reveal connectivity patterns among some particular regions.

Another benefit emanating from the new JD-based method is that it effectively circumvents some computationally heavy operations. For example, large matrix-matrix multiplications in CPD are no longer—unlike the existing tensor-based method [28]. Hence, the JD-based approach has a better potential to deal with large-scale networks. Furthermore, JD can naturally incorporate prior knowledge on networks, e.g., sparsity and local smoothness. This is important, since real-world network data are often noisy, and using prior information as regularization terms and/or constraints may help enhance the performance of network inference.

JD is a well appreciated tool in computational linear algebra, and finds applications in a number of signal processing applications, prominently in blind speech and audio separation [6], [22], [32]–[34], [36], [37]. General-purpose off-the-shelf JD algorithms can be also employed in our present context. To better serve the goals of network topology inference however, three new customized JD algorithms will be proposed: (i) JD-based SEM for general topology identification from second-order statistics; (ii) sparse-JD based SEM (S-JDSEM) designed for sparse connectivity networks; and, (iii) robust (R)-JDSEM to identify the network structure in the presence of abnormal nodes or outliers. These variants can be flexibly applied depending on the prior information available about the network structure or the noise model. Compared with our conference precursor [27], the present work is capable of exploring the prior structural information such as sparsity, or presence of outliers. In addition, identifiability analysis will be carried out to show that the JD model is capable of identifying the underlying network structure under mild conditions.

The rest of this paper is organized as follows. Preliminaries and a formal statement of the problem are in Section II, while Section III deals with JD of tensor correlation slabs, and introduces three variants of the basic JD solver suitable for different scenarios. Section IV presents identifiability results for the proposed framework. Finally, corroborating numerical tests on both synthetic and real data are presented in Section V, while concluding remarks along with a discussion of ongoing and future directions are the subject of Section VI.

Notation. Bold uppercase (lowercase) letters will denote matrices (column vectors), while operators $(\cdot)^\top$ and $\lambda_{\max}(\cdot)$ will stand for matrix transposition and maximum eigenvalue, respectively. The identity matrix will be denoted by $\mathbf{I}$, while ℓ_p and Frobenius norms will be denoted by $\|.\|_p$ and $\|.\|_F$, respectively. The operator vec$(.)$ will vertically stack columns of its matrix argument, to form a vector. Finally, $\mathbf{A}\otimes\mathbf{B}$ will stand for the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$, while $\mathbf{A}\odot\mathbf{B}$ will denote their Khatri-Rao product, namely, $\mathbf{A}\odot\mathbf{B}:=[\mathbf{a}_1\otimes\mathbf{b}_1,\ldots\mathbf{a}_N\otimes\mathbf{b}_N]$, where $\mathbf{A}:=[\mathbf{a}_1,\ldots,\mathbf{a}_N]$ and $\mathbf{B}:=[\mathbf{b}_1,\ldots,\mathbf{b}_N]$.

II. Preliminaries and Problem Statement

In the present section, we will first introduce SEM-based topology identification [14]. Following this, we will outline tensor-based topology inference, before proceeding to our novel JD-based approach.

A. Background

Consider a graph denoted as $\mathcal{G}(\mathcal{V},\mathcal{E})$ with N nodes and its adjacency matrix $\mathbf{A}\in\mathbb{R}^{N\times N}$ having (i,j)th entry a_{ij} that is nonzero if and only if there is an edge linking node j to node i. We allow $\mathcal{G}$ to be directed, meaning $\mathbf{A}$ can be asymmetric $(\mathbf{A}\neq\mathbf{A}^\top)$. Suppose this graph is an abstraction of a complex system with measurable inputs and outputs that propagate over the network following directed links. Let x_{it} denote the input to node i during time slot t, and y_{it} the t-th observation of the propagating process measured at node i. The signals y_{it} and x_{it} can have different physical meanings depending on the application. In the context of brain networks for example, y_{it} represents the t-th time sample of a certain measuring scheme (e.g., functional magnetic resonance imaging (fMRI)) at region i, while x_{it} represents a stimulus exciting a specific region of the brain. In financial networks, y_{it} denotes the closing price of stock i on the t-th day, while x_{it} represents the money invested to a specific stock on day t.

1) SEM-based approach: Here, we generally postulate that y_{it} depends on two classes of variables, namely: i) measurements of the diffusing process $\{y_{jt}\}_{j\neq i}$ (a.k.a. *endogenous* variables); and ii) external inputs x_{it} (*exogenous* variables). SEM-based approaches posit that y_{it} depends linearly on both $\{y_{jt}\}_{j\neq i}$ and x_{it}; that is,

$$y_{it}=\sum_{j\neq i}a_{ij}y_{jt}+b_{ii}x_{it}+e_{it} \tag{1}$$

where e_{it} denotes the term that captures unmodeled dynamics. The coefficients $\{a_{ij}\}$ and $\{b_{ii}\}$ are unknown, and $a_{ij}\neq 0$ signifies that a directed edge from j to i is present. Collecting nodal measurements $\mathbf{y}_t:=[y_{1t}\ldots y_{Nt}]^\top$, and $\mathbf{x}_t:=[x_{1t}\ldots x_{Nt}]^\top$ per slot t, the noise-free version of (1) can be written as

$$\mathbf{y}_t=\mathbf{A}\mathbf{y}_t+\mathbf{B}\mathbf{x}_t \tag{2}$$

where $[\mathbf{A}]_{ii}=0$ and $\mathbf{B}:=\text{Diag}(b_{11},\ldots,b_{NN})$ denotes a diagonal matrix with $b_{ii}\neq 0$, for $i=1,\ldots,N$ as its diagonal entries. Classic SEM-based approaches assume that $\{\mathbf{x}_t,\mathbf{y}_t\}_{t=1}^T$ are available. If $\text{rank}(\mathbf{Y})=N$, then using that $a_{ii}=0$ and $b_{ii}\neq 0\ \forall i$, matrices $\mathbf{A}$ and $\mathbf{B}$ turn out to be identifiable as the solution of the linear system of equations [5], [8]

$$(\mathbf{I}-\mathbf{A})\mathbf{Y}=\mathbf{B}\mathbf{X}$$

where $\mathbf{X}:=[\mathbf{x}_1,\ldots,\mathbf{x}_T]$, and $\mathbf{Y}:=[\mathbf{y}_1,\ldots,\mathbf{y}_T]$.

The conventional SEM-based approach solves the following problem: Given $\mathbf{X}$ and $\mathbf{Y}$, find $\mathbf{A}$ by solving the problem $\min_{\mathbf{A},\mathbf{B}}\|\mathbf{Y}-\mathbf{A}\mathbf{Y}-\mathbf{B}\mathbf{X}\|_F^2$. Sparsity of the network connectivity was leveraged by [5] and [2] in order to improve identification performance. Note that the conventional SEM assumes availability of nodal measurements.

2) Tensor-based approach: Here, the exogenous inputs across nodes are not measurable, but are assumed to satisfy the following.

(as0) Exogenous inputs $\{\mathbf{x}_t^{(m)}\}$ are *piecewise wide-sense stationary* over time segments $t\in[\tau_m,\tau_{m+1}-1],m=1,\ldots,M+1$, each with a fixed correlation matrix $\mathbf{R}_m^x:=$

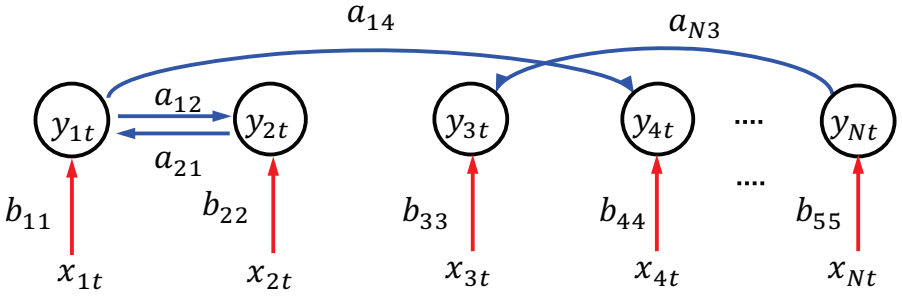


Fig. 1: An N-node directed network (blue links), with the t-th samples of endogenous measurements per node. SEMs explicitly account also for exogenous inputs (red arrows), upon which endogenous variables may depend [12].

$\mathbb{E}\{\mathbf{x}_t^{(m)}(\mathbf{x}_t^{(m)})^\top\}$;
(as1) Entries of $\mathbf{x}_t$ are zero mean and spatially uncorrelated; that is, $\mathbb{E}\{x_{it}x_{jt}\}=0, \forall i\neq j$; and
(as2) Matrix $(\mathbf{I}-\mathbf{A})$ is invertible;

Under (as2), the model in (2) can be re-written as

$$\mathbf{y}_t=(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\mathbf{x}_t=\boldsymbol{\mathcal{A}}\mathbf{x}_t \tag{3}$$

where $\boldsymbol{\mathcal{A}}:=(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}$, and superscript (m) has been dropped with the understanding that t stays within one segment, and thus (3) holds $\forall m$. The *per segment* correlation matrix $\mathbf{R}_m^y:=\mathbb{E}\{\mathbf{y}_t\mathbf{y}_t^\top\}$ is thus given by [cf. (3) and (as0)]

$$\mathbf{R}_m^y=\boldsymbol{\mathcal{A}}\mathbf{R}_m^x\boldsymbol{\mathcal{A}}^\top,\quad t\in[\tau_m,\tau_{m+1}-1]. \tag{4}$$

Under (as1), one can express (4) as the weighted sum of rank-one matrices as

$$\mathbf{R}_m^y=\boldsymbol{\mathcal{A}}\text{Diag}(\boldsymbol{\rho}_m^x)\boldsymbol{\mathcal{A}}^\top=\sum_{i=1}^N\rho_{mi}^x\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top \tag{5}$$

where $\boldsymbol{\alpha}_i$ denotes the ith column of $\boldsymbol{\mathcal{A}}$, and $\boldsymbol{\rho}_m^x:=[\rho_{m1}^x\ldots\rho_{mN}^x]^\top$, with $\rho_{mi}^x:=\mathbb{E}(x_{it}^2)$, for $t\in[\tau_m,\tau_{m+1}-1]$.

Consider the three-way tensor $\underline{\mathbf{R}}^y\in\mathbb{R}^{N\times N\times M}$, constructed by setting the m-th slice $[\underline{\mathbf{R}}^y]_{:,:,m}=\mathbf{R}_m^y$. Letting $\alpha_{ji}\beta_{ki}\gamma_{li}$ denote the (j,k,l) entry of the tensor outer product $\boldsymbol{\alpha}_i\circ\boldsymbol{\beta}_i\circ\boldsymbol{\gamma}_i$, where $\alpha_{ji}:=[\boldsymbol{\alpha}_i]_j$ (resp. β_{ik} and γ_{il}), it turns out that $\underline{\mathbf{R}}^y$ can be written as

$$\underline{\mathbf{R}}^y=\sum_{i=1}^N\boldsymbol{\alpha}_i\circ\boldsymbol{\alpha}_i\circ\mathbf{r}_i^x \tag{6}$$

with entry (j,k,l) given by

$$[\underline{\mathbf{R}}^y]_{jkl}=\sum_{i=1}^N\alpha_{ji}\alpha_{ki}r_{li}^x \tag{7}$$

where $\mathbf{r}_i^x:=[\rho_{1i}^x\ldots\rho_{Mi}^x]^\top$. Interestingly, (6) amounts to the partially symmetric CPD of $\underline{\mathbf{R}}^y$ into factor matrices $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{A}}$, and $\mathbf{R}^x:=[\mathbf{r}_1^x\ldots\mathbf{r}_N^x]\in\mathbb{R}^{M\times N}$; see e.g., [16], [28]. Although $\mathbf{R}_m^y$ is generally unknown, it can be readily estimated using sample averaging of endogenous measurements, as

$$\widehat{\mathbf{R}}_m^y=\frac{1}{\tau_{m+1}-\tau_m}\sum_{t=\tau_m}^{\tau_{m+1}-1}\mathbf{y}_t\mathbf{y}_t^\top,\quad m=1,\ldots,M\,. \tag{8}$$

Supposing that second-order statistics of the network processes are only available, the goal of the tensor based approach is to find $\mathbf{A}$ and $\mathbf{B}$ given tensor $\hat{\underline{\mathbf{R}}}^y$ with $\{\widehat{\mathbf{R}}_m^y\}$ as its m th slab.

Specifically, relying on this three-way tensor constructed from second-order statistics of the nodal measurements, [27] leverages CPD to identify the latent topology $\boldsymbol{\mathcal{A}}$; e.g., via alternating least-squares (ALS) iterations. Under (as2), it is possible to recover $\mathbf{A}$, once $\boldsymbol{\mathcal{A}}$ has been found as

$$\mathbf{A}=\mathbf{I}-\left(\text{Diag}(\boldsymbol{\mathcal{A}}^{-1})\right)^{-1}\boldsymbol{\mathcal{A}}^{-1}. \tag{9}$$

Despite the fact the CPD based SEM (CPSEM) is capable of identifying the network structure without exact information about the exogenous input of each node, it still faces several important challenges:

c1. The approach in [27] relies on the ALS algorithm to perform the CPD for estimating $\mathcal{A}$, which leads to serious scalability issues. The latent factor $\mathcal{A}$ in CPSEM has size $N\times N$, and the tensor rank is N. Hence, the key ALS operation involving the *matricized tensor times Khatri-Rao product* (MTTKRP) costs $O(N^4M)$ flops; see [30]. Note that N can be millions in a complex network, and MTTKRP is carried out three times in *each iteration* of ALS for this particular problem.
c2. Matrix $\boldsymbol{\mathcal{A}}$ does not capture the structure of $\mathbf{A}$, e.g., sparsity, since it is obtained by first inverting $\mathbf{A}$; and,
c3. Even though exact information of $\mathbf{x}_t$ is no longer required for the tensor-based algorithm to recover the network topology, it is still necessary to know the second-order statistics $\mathbf{R}_x$ [27], which may not be available in certain applications. To circumvent these challenges, we will show how to leverage JD of the tensor slices.

B. Problem Setup

Different from conventional SEM settings, where exogenous inputs $\{\mathbf{x}_t\}_{t=1}^T$ are assumed known, we consider here that such information is not available. Instead, we assume that the connectivity patterns of a small subset of nodes, referred to as *anchor nodes*, are known. The problem statement can now be formally stated as follows.

Problem statement: Given second-order statistics of $\{\mathbf{y}_t\}_{t=1}^T$, and the connectivity pattern of a few anchor nodes, the goal is to recover the underlying directed network topology $\mathbf{A}$ while exploiting the possibly available network structure, e.g. edge sparsity.

A couple of examples to motivate this specific problem setup are due next.

Example 1. In brain networks, although the stimuli $\{\mathbf{x}_t\}$ are not easy to measure, the interaction among some regions of the brain are well understood, e.g., based on biological connectivity using diffusion tensor imaging [18], or prior domain research results. Such regions can be viewed as the anchor nodes.

Example 2. In a financial stock market, where each stock denotes a node, y_{it} represents the stock price of stock i at time t, and the exogenous input x_{it} is the money invested into stock i at time t. While stock prices can be observable, the investment over time may not be accessible due to privacy concerns. However, the influence of some stocks can be forecast based on historical data; see e.g., [14].

III. JD-based Topology Identification

The topology inference problem will be formulated here as a JD task using the notion of anchor nodes. The resultant solver will be broadened to account for sparsity, and also gain resilience to outliers.

A. Topology inference with anchor nodes

To begin, consider rewriting (2) as

$$(\mathbf{I}-\mathbf{A})\mathbf{y}_t=\mathbf{B}\mathbf{x}_t \tag{10}$$

and with $\mathbf{H}:=\mathbf{I}-\mathbf{A}$, write the *per-segment* correlation matrix $\mathbf{R}_m^y:=\mathbb{E}\{\mathbf{y}_t\mathbf{y}_t^\top\}$, as

$$\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top=\mathbf{B}\mathbf{R}_m^x\mathbf{B}^\top \tag{11}$$

where $\mathbf{B}$ and $\mathbf{R}_m^x$ are unknown diagonal matrices as per (as1). Clearly, (11) implies that $\mathbf{R}_m^y,\ m=1,\ldots,M$ are jointly diagonalizable by $\mathbf{H}$. In this noise-free setup, the latter yields

$$\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top=\mathrm{Diag}(\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top),\qquad m=1,\ldots,M. \tag{12}$$

With h_{ij} denoting the (i,j)th entry of $\mathbf{H}$, it is easy to see that h_{ij} satisfies

$$h_{ii}=1,\qquad \forall i \tag{13a}$$
$$h_{ij}=-a_{ij}\quad \forall i\neq j \tag{13b}$$

which suggests identifying $\mathbf{H}$ by solving

$$\min_{\mathbf{H}}\sum_{m=1}^M\|\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top-\mathrm{Diag}(\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top)\|_F^2$$
$$\text{s.t.}\quad h_{ii}=1,\quad \forall i. \tag{14}$$

Altough intuitively pleasing, further reflection on (14) reveals that if $\mathbf{H}^*$ is an optimal solution to (14), then any

$$\bar{\mathbf{H}}=\mathbf{H}^*\mathbf{\Pi}\mathbf{\Lambda}$$

is also an optimal solution, where $\mathbf{\Pi}$ denotes a permutation matrix, and $\mathbf{\Lambda}$ is a nonsingular diagonal scaling matrix. This is because column permutation and scaling ambiguities are intrinsic to JD [6] (as in tensor and matrix decomposition), and cannot be removed without extra information.

Nonetheless, permutation and scaling ambiguities are not tolerable in network topology identification, since they make isomorphic graphs indistinguishable. In order to resolve the permutation ambiguity, [28] relies on additional information about the second-order statistics of the exogenous variables, that is $\mathbf{R}^x$. This idea can also be used in the JD-based approach. However, the present paper introduces an alternative for ambiguity removal. To be specific, this work will utilize the notion of *anchor nodes* to pin down the final estimate of the network topology. Here, anchor refers to a node whose connectivity patterns are known *a priori* (cf. Examples 1-2).

Specifically, we introduce the following.

(as3) A subset of $[\mathbf{A}]_{ij}$ entries with $(i,j)\in\Omega$ is known.

Incorporating the known entries as constraints into (14) yields the constrained least-squares problem

$$\min_{\mathbf{H}}\sum_{m=1}^M\|\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top-\mathrm{Diag}(\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top)\|_F^2$$
$$\text{s.t.}\quad h_{ii}=1\ \ \forall i,\qquad h_{ij}=-a_{ij}\ \ \forall(i,j)\in\Omega. \tag{15}$$

Through this formulation, it is possible under (as0), (as1) and (as3) to estimate $\mathbf{H}$, and then the adjacency matrix $\mathbf{A}$. Detailed identifiability analysis will be provided in Section IV.

Note that (15) is not a standard JD formulation because it includes special constraints. Nevertheless, it can be tackled following ideas from classic JD algorithms, e.g., via block coordinate descent (BCD). Note that BCD converges to a critical point of the optimization problem under certain conditions—e.g., when the block subproblems are strictly convex; see [23], [35]. A simple and intuitive algorithm can be derived, starting with $\mathbf{H}:=[\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_N]^\top$, where $\mathbf{h}_n^\top$ denotes the nth row of $\mathbf{H}$. Supposing $\mathbf{h}_1,\ldots,\mathbf{h}_{n-1},\mathbf{h}_{n+1},\ldots,\mathbf{h}_N$ are fixed, the subproblem w.r.t. $\mathbf{h}_n$ can be written as

$$\min_{\mathbf{h}_n}\|\mathbf{W}_n\mathbf{h}_n\|_2^2$$
$$\text{s.t.}\quad h_{nn}=1,\qquad h_{nj}=-a_{nj}\ \ \forall(n,j)\in\Omega \tag{16}$$

where

$$\begin{aligned}\mathbf{W}_n&:=[\mathbf{Q}_{n1}^\top\ldots\mathbf{Q}_{nM}^\top]^\top\\ \mathbf{Q}_{nm}&:=\mathbf{H}_{-n}\mathbf{R}_m^y\\ \mathbf{H}_{-n}&:=[\mathbf{h}_1\ldots\mathbf{h}_{n-1}\mathbf{h}_{n+1}\ldots\mathbf{h}_N]^\top.\end{aligned} \tag{17}$$

Problem (16) is an equality-constrained quadratic program, which can be solved in closed form. To see this, let $\Omega_n:=\{j:(n,j)\in\Omega\}$ denote the set that consists of column indices of the known entries in the nth row of $\mathbf{A}$. In addition, define $\bar{\mathbf{W}}_n$ as the submatrix of $\mathbf{W}$ constructed by the columns of $\mathbf{W}_n$ with indices not in Ω_n. Use $\bar{\mathbf{h}}_n$ to represent the subvector of $\mathbf{h}_n$ collecting entries indexed by Ω_n. One can now re-write (16) as

$$\min_{\bar{\mathbf{h}}_n}\|\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n+\boldsymbol{\epsilon}_n\|_2^2 \tag{18}$$

where

$$\boldsymbol{\epsilon}_n:=\sum_{j\in\Omega_n}\mathbf{w}_jh_{nj}+\mathbf{w}_n \tag{19}$$

with $\mathbf{w}_n$ denoting the nth column of $\mathbf{W}$ that is known per subproblem n. It is clear that for each sub-problem, the closed-

Algorithm 1 Topology inference via joint diagonalization

Input: $\mathbf{R}_{\Omega}^{x}$, $\{\mathbf{y}_t\}$, M, η

S1. Tensor construction:

Set m-th frontal slice of $\underline{\mathbf{R}}^{y} \in \mathbb{R}^{N\times N\times M}$ to $\widehat{\mathbf{R}}_{m}^{y} = \frac{1}{\tau_{m+1}-\tau_m}\sum_{t=\tau_m}^{\tau_{m+1}-1}\mathbf{y}_t\mathbf{y}_t^{\top},\ m=1,\ldots,M$

S2. Tensor decomposition via joint diagonalization:

Estimate $\hat{\mathbf{H}}$ by solving (15), (21) or (27).

S3. SEM estimates for topology inference:

$\hat{\mathbf{A}} = \mathbf{I} - \hat{\mathbf{H}}$

S4. Edge identification:

$[\hat{\mathbf{A}}]_{ij} \neq 0$ if $[\hat{\mathbf{A}}]_{ij} > \eta$, otherwise $[\hat{\mathbf{A}}]_{ij} = 0$, $\forall(i,j)$

form solution is readily obtained as

$$\bar{\mathbf{h}}_n = -\bar{\mathbf{W}}_n^{\dagger}\boldsymbol{\epsilon}_n \tag{20}$$

where $\bar{\mathbf{W}}_n^{\dagger} := (\bar{\mathbf{W}}_n^{\top}\bar{\mathbf{W}}_n)^{-1}\bar{\mathbf{W}}_n$ is the pseudo-inverse of $\bar{\mathbf{W}}_n$.

Remark 1: The per-iteration core complexity of this algorithm mainly comes from (20), which is of the order $\mathcal{O}(N^3)$. This is already at least two orders of magnitude lighter compared to $\mathcal{O}(N^4M)$ that was needed in CPSEM. Note that the per-iteration complexity can be further reduced, leveraging first-order optimization and inexact BCD—which will be introduced in the ensuing subsections together with topology structure-aware variants of the JD formulation.

B. Topology inference via sparse JD

To further capitalize on the JD formulation, the present section studies the case where the network of interest exhibits edge sparsity. Such property is pervasive in real-world networks where each node only has a small number of neighbors compared with the total number of nodes in the network. It will be shown that the novel JD-based approach can flexibly incorporate such structural information without incurring high computational complexity.

Unlike the previous tensor-based approach which requires matrix inversion to recover the network topology [c.f. (9)], factor $\mathbf{H}$ in the JDSEM algorithm naturally inherits the sparsity pattern of $\mathbf{A}$ [cf (13)]. Therefore, if the network is sparse, only a small subset of $\{h_{ij}\}$s will be nonzero, and the nonzero positions correspond to the presence of an edge, meaning $a_{ij} \neq 0$. Such an observation leads to the following sparsity regularized criterion

$$\begin{aligned}\min_{\mathbf{H}}\ &\frac{1}{2}\sum_{m=1}^{M}\|\mathbf{H}\mathbf{R}_m^{y}\mathbf{H}^{\top} - \mathrm{Diag}(\mathbf{H}\mathbf{R}_m^{y}\mathbf{H}^{\top})\|_F^2 + \lambda\|\mathbf{H}\|_1\\ \text{s.t.}\ \ &h_{ii} = 1\ \ \forall i, \qquad h_{ij} = -a_{ij}\ \ \forall(i,j)\in\Omega\end{aligned} \tag{21}$$

where $\|\mathbf{H}\|_1 := \sum_{ij}|h_{ij}|$. Similar to (15), a BCD iteration is applied to solve (21). Clearly, it boils down to the per-row subproblem [cf. (17)]

$$\begin{aligned}\min_{\mathbf{h}_n}\ &\frac{1}{2}\|\mathbf{W}_n\mathbf{h}_n\|_2^2 + \lambda\|\mathbf{h}_n\|_1\\ \text{s.t.}\ \ &h_{nn} = 1, \qquad h_{nj} = -a_{nj}\ \ \forall j\in\Omega_n\end{aligned} \tag{22}$$

where $\lambda > 0$ is the regularization parameter, and $\mathbf{W}_n$ is as in (17). The problem in (22) is convex but nonsmooth, and several off-the-shelf convex optimization solvers can be employed; e.g., proximal splitting such as proximal gradient descent iterations [9], or, the alternating direction method of multipliers (ADMM) [7], [25].

Although ADMM is a viable solution, it may involve large matrix inversion in its updates that incurs $\mathcal{O}(N^3)$ complexity. Instead, the proximal gradient (PG) method is adopted here. Specifically, by eliminating the constraints as before, (22) can be written as the following unconstrained problem

$$\min_{\bar{\mathbf{h}}_n}\frac{1}{2}\|\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n\|_2^2 + \lambda\|\bar{\mathbf{h}}_n\|_1. \tag{23}$$

The PG algorithm results in the closed-form update

$$\bar{\mathbf{h}}_n^{k} = \mathcal{P}_{\lambda/L_f}(\bar{\mathbf{h}}_n^{k-1} - \nabla f(\bar{\mathbf{h}}_n^{k-1})/L_f) \tag{24}$$

with $\mathcal{P}_z(.)$ denoting entry-wise soft thresholding given by

$$\mathcal{P}_{\rho}(z) := \frac{z}{|z|}\max(|z|-\rho, 0). \tag{25}$$

With $\nabla f(\bar{\mathbf{h}}_n^{k-1})$ denoting the gradient of the continuously differentiable $f(\bar{\mathbf{h}}_n) := \frac{1}{2}\|\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n\|_2^2$, one arrives at

$$\nabla f(\bar{\mathbf{h}}_n) = \bar{\mathbf{W}}_n^{\top}(\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n) \tag{26}$$

which is Lipschitz continuous with constant $L_f := \lambda_{\min}(\bar{\mathbf{W}}_n^{\top}\bar{\mathbf{W}}_n)$.

Remark 2: The PG algorithm complexity largely depends on that for computing the gradient in (26), which no longer requires computing a matrix inversion and incurs complexity of $\mathcal{O}(N^2M)$.

C. Robust sparse JDSEM

In several challenging scenarios, the local covariance matrices $\{\mathbf{R}_y^m\}$ may not be jointly diagonalizable due to the presence of outliers. This is a common challenge for real world networks, due to adversarial nodes that generate abnormal signals. In social networks for instance, spammers may be present sending out malicious emails that can be viewed as outliers. Another example could be malfunctioning nodes in power networks or abnormal regions of interest in brain networks. To cope with such potential outliers, a robust version of JDSEM (R-JDSEM) is developed in the present subsection. The outliers present can be modeled as sparse noise components in the matrix of second-order statistics that cannot be jointly diagonalizable.

In order to account for the potential outliers, the ℓ_1-norm fitting term can replace the LS one in (21), which leads to the ℓ_1-regularized minimum absolute value minimization

$$\begin{aligned}\min_{\mathbf{H}}\ &\sum_{m=1}^{M}\|\mathbf{H}\mathbf{R}_m^{y}\mathbf{H}^{\top} - \mathrm{Diag}(\mathbf{H}\mathbf{R}_m^{y}\mathbf{H}^{\top})\|_1 + \lambda\|\mathbf{H}\|_1\\ \text{s.t.}\ \ &h_{ii} = 1\ \ \forall i, \qquad h_{ij} = -a_{ij}\ \ \forall(i,j)\in\Omega\end{aligned} \tag{27}$$

and the subproblem per block $\mathbf{h}_n$ in each BCD iteration can

be written as

$$\min_{\mathbf{h}_n} \|\mathbf{W}_n\mathbf{h}_n\|_1 + \gamma\|\mathbf{h}_n\|_1$$
$$\text{s.t.} \quad h_{nn} = 1, \qquad h_{nj} = -a_{nj} \quad \forall (n,j) \in \Omega\,. \tag{28}$$

Again, (28) is a convex and non-smooth problem that can be solved via proximal splitting methods such as ADMM, or BCD iterations; see Appendix A for detailed derivation. The per-iteration computational complexity of the R-JDSEM solver here is $\mathcal{O}(N^3)$.

IV. Identifiability analysis

So far, we have seen that the directed network topology identification task can be cast into a joint diagonalization problem. The present section aims at providing identifiability conditions for the proposed model, which will be analyzed along the lines of exact JD of tensors. As usual, identifiability of the network topology will be analyzed in the noiseless case, where the local covariance matrix can be exactly diagonalized. In this case, the optimal solution $\mathbf{H} = \mathbf{I} - \mathbf{A}$ satisfies that

$$\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top = \text{Diag}(\mathbf{H}\mathbf{R}_m^y\mathbf{H}^\top) \tag{29}$$
$$\text{s.t.} \quad h_{ii} = 1 \quad \forall i, \qquad h_{ij} = -a_{ij} \quad \forall (i,j) \in \Omega.$$

To proceed, we will need a couple of definitions.

Definition 1. The *Kruskal rank* of a matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ (denoted hereafter as $\text{kr}(\mathbf{Z})$) is the maximum number k of *any* collection of k columns of $\mathbf{Z}$ forming a full-rank submatrix.

Definition 2. *Essential* uniqueness of a tensor factorization refers to uniqueness up to scale and permutation ambiguities. Let $(\mathbf{U},\mathbf{V},\mathbf{W})$ denote the PARAFAC factors obtained by decomposing a three-way tensor $\underline{\mathbf{Z}}$ into K rank-one tensors. If an alternative triplet $(\bar{\mathbf{U}},\bar{\mathbf{V}},\bar{\mathbf{W}})$ satisfies the PARAFAC decomposition of $\underline{\mathbf{Z}}$, there must exist a permutation matrix $\mathbf{\Pi}$, and diagonal matrices $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$, so that $\mathbf{\Lambda}_1\mathbf{\Lambda}_2\mathbf{\Lambda}_3 = \mathbf{I}$, $\bar{\mathbf{U}} = \mathbf{U}\mathbf{\Pi}\mathbf{\Lambda}_1$, $\bar{\mathbf{V}} = \mathbf{V}\mathbf{\Pi}\mathbf{\Lambda}_2$, and $\bar{\mathbf{W}} = \mathbf{W}\mathbf{\Pi}\mathbf{\Lambda}_3$.

Without extra prior information, it is well known that only essential uniqueness can be guaranteed when $\mathbf{R}_x$ is fully available and $\text{kr}(\mathbf{R}^x) \geq 2$; see [31] and [17] for further details. In the present paper, the goal is to analyze the uniqueness in the presence of anchor nodes with known connectivity patterns. The following result asserts how many anchor nodes suffice to guarantee JD-based reconstruction of the topology.

Theorem 1: *If $\mathbf{x}_t$ and $\mathbf{y}_t$ adhere to the SEM in* (2)*, along with (as0) and (as1), for $t = 1,\ldots,$ and if $\text{kr}(\mathbf{R}^x) \geq 2$, then $\mathbf{A}$ can be uniquely identified if the nonzero entries of $\mathbf{A}$ are randomly drawn from a continuous distribution, and at least two off-diagonal nonzero entries are known a priori in each column of $\mathbf{A}$.*

Proof: It has been shown in [1] that uniqueness via exact JD can be viewed as a special case of tensor factorization, and the Kruskal's condition is sufficient for essential uniqueness of the estimated factors. Hence, recalling Kruskal's condition [17], it can be readily deduced that if the Kruskal ranks of $\mathbf{H}$ and $\mathbf{R}_x$ satisfy

$$\text{kr}(\mathbf{R}_x) + 2\text{kr}(\boldsymbol{\mathcal{A}}) \geq 2N + 2 \tag{30}$$

essential uniqueness is guaranteed, meaning $\hat{\mathbf{H}}$ satisfies

$$\hat{\mathbf{H}} = \mathbf{H}\mathbf{\Lambda}\mathbf{\Pi} \tag{31}$$

where $\mathbf{\Pi}$ is a permutation matrix and $\mathbf{\Lambda}$ a diagonal scaling matrix. Based on (as1) and (as2), $\boldsymbol{\mathcal{A}} := (\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is invertible, meaning that $\text{kr}(\boldsymbol{\mathcal{A}}) = N$. Hence, condition (30) is satisfied if $\text{kr}(\mathbf{R}_x) \geq 2$.

Let N_j denote the number of known entries in the jth column of $\mathbf{A}$, and let $\mathbf{h}_\Omega^j \in \mathbb{R}^{N_j+1}$ represent the sub-vector of $\mathbf{h}_j$ formed by the a priori known entries, while $\hat{\mathbf{h}}_\Omega^j$ is the corresponding sub-vector of $\hat{\mathbf{h}}_j$. The constraints in (29) related to the jth column can be written as

$$\hat{\mathbf{h}}_\Omega^j = \mathbf{h}_\Omega^j. \tag{32}$$

Upon combining (31) with (32), we arrive at

$$\mathbf{h}_\Omega^j - \mathbf{H}_\Omega^j\mathbf{p}_j = \mathbf{0}_{(N_j+1)\times 1} \tag{33}$$

where $\mathbf{H}_\Omega^j$ denotes the sub-matrix of $\mathbf{H}$ formed by the rows corresponding to the available entries in $\mathbf{h}_j^\Omega$, and $\mathbf{p}_j \in \mathbb{R}^N$ represents the jth column of $\mathbf{P} := \mathbf{\Lambda}\mathbf{\Pi}$. Since $\mathbf{\Pi}$ is a permutation matrix, each constituent column in $\mathbf{\Pi}$ comprises zeros with the exception of a single entry set to one. Letting π_{ij} denote the (i,j)-th entry of $\mathbf{\Pi}$, assume without loss of generality that $\pi_{ij} = 1$ and $\pi_{kj} = 0$, $\forall k \neq i$. Consequently, with $\mathbf{p}_j \in \mathbb{R}^N$ representing column j of $\mathbf{P} := \mathbf{\Pi}\mathbf{\Lambda}$, one can equivalently write

$$\mathbf{p}_j = [0,\ldots,0,\underbrace{\lambda_i\pi_{ij}}_{\text{entry } i},0,\ldots,0]^\top \tag{34}$$

where λ_j denotes the j-th diagonal entry of $\mathbf{\Lambda}_3$. Combining (34) with (33), one obtains

$$\mathbf{h}_\Omega^j = \lambda_i\pi_{ij}\mathbf{h}_\Omega^i. \tag{35}$$

For $i \neq j$, (35) implies that $\mathbf{h}_\Omega^i$ and $\mathbf{h}_\Omega^j$ are linearly dependent. On the other hand, if $N_j \geq 1$, we have that the size of $\mathbf{h}_\Omega^j$ is at least $N_j + 1 \geq 2$. However, since nonzero entries of $\mathbf{A}$ are drawn from a continuous distribution, any two columns of $\mathbf{H}_\Omega^j$ are linearly independent with probability 1, which contradicts (35). It further holds that $\mathbf{h}_\Omega^j \neq \mathbf{0}$ with probability 1 for all j, meaning $\lambda_i \neq 0$. Therefore, the trivial solution of $\mathbf{\Lambda} = \mathbf{0}$ can be avoided. To this end, for (35) to hold, it is necessary that $i = j$, which is equivalent to requiring $\pi_{jj} = 1$ and $\lambda_j = 1$. Since this holds for any j, one deduces that

$$\mathbf{\Pi} = \mathbf{I}, \quad \mathbf{\Lambda} = \mathbf{I}. \tag{36}$$

Combining with (31), yields $\hat{\mathbf{H}} = \mathbf{H}$, and also completes the proof of Theorem 1. ■

Note that the assumption requiring that at least two entries are known *a priori* per column of $\mathbf{A}$ can be satisfied if there exist two anchor nodes whose *in-links* have been obtained, e.g., via prior domain study. Having two anchor nodes whose in-links are known is a mild assumption in many applications. For example, in brain network analysis, there might be already established connectivity patterns of some specific areas, and thus such information can provide the needed anchor nodes.

Remark 3 (Comparison with [28]): Different from the

identifiability results in [28], Theorem 1 here asserts that exact identification of network structure is possible, even when no information on the $\mathbf{x}_t$ statistics is available, as long as the in-links of certain nodes are known *a priori*. Thus, the novel methods here offer a useful alternative to existing approaches, in cases when one has no access to statistics of external inputs. Also, unlike [28], it turns out that our joint-diagonalization based method does not require the matrix $(\mathbf{I}-\mathbf{A})$ to be invertible [c.f. (as2)], because we directly find $\mathbf{H}=\mathbf{I}-\mathbf{A}$. Moreover, without the need of matrix inversion after the network identification stage, it will be shown in Section V that the novel JD-based methods are empirically more robust to different noise settings as well as different data models.

V. Numerical Tests

In this section, the performance of the proposed algorithms is tested on both synthetic and real datasets.

A. Synthetic data tests with Gaussian noise

Data generation. A *Kronecker random graph* comprising $N=64$ nodes was generated from a prescribed "seed matrix"

$$\mathbf{S}_0 := \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

in order to obtain a binary-valued 64×64 matrix using Kronecker product operations to find $\mathbf{S}=\mathbf{S}_0\otimes\mathbf{S}_0\otimes\mathbf{S}_0$; see also [19]. With the binary matrix $\mathbf{S}$ setting the positions of zero and nonzero entries of the topology, a Kronecker graph with adjacency matrix $\mathbf{A}$ was then constructed by randomly sampling each entry from a uniform distribution with $a_{ij}\sim \text{Unif}(0.2s_{ij}, 0.5s_{ij})$. To generate endogenous measurements, the observation horizon was set to $T=ML$ time slots, which were partitioned into M windows of fixed length L, using pre-selected boundaries $\{\tau_m\}_{m=1}^{M+1}$ with $\tau_1=1$ and $L:=\tau_{m+1}-\tau_m$, for several values of L and M. Per $t\in[\tau_m,\tau_{m+1}-1]$, exogenous inputs were sampled as $\mathbf{x}_t\sim\mathcal{N}(\mathbf{0},\sigma_m^2\mathbf{I})$, with $\{\sigma_m\}_{m=1}^M$ set to M distinct values. With $\mathbf{e}_t$ sampled i.i.d. from $\mathcal{N}(\mathbf{0},\sigma_e^2\mathbf{I})$, vector $\mathbf{y}_t$ was generated using the SEM, that is, $\mathbf{y}_t=(\mathbf{I}-\mathbf{A})^{-1}(\mathbf{B}\mathbf{x}_t+\mathbf{e}_t)$, where $\mathbf{B}$ is a diagonal matrix with diagonal entries $[\mathbf{B}]_{jj}$ drawn uniformly from the interval $[2,3]$.

Test results. The performance of JDSEM is first tested versus different window lengths, and different numbers of time segments or measurements. Figure 2 illustrates the observed error performance in terms of NMSE $:=\|\mathbf{A}-\hat{\mathbf{A}}\|_F^2/\|\mathbf{A}\|_F^2$ over several window lengths (L) when $N_a=5$ anchor nodes are present. In addition, we test the performance when the identifiability condition in Theorem 1 is not satisfied. Specifically, given the generated random graphs, we randomly select one node, and remove all edges connecting the selected node to other nodes, except one. The NMSE performance in this case is illustrated in Figure 3. It can be observed that, compared with Figure 2, the NMSE obtained is slightly higher, but not significantly so. This shows that the algorithm can still provide reliable performance even if the identifiability conditions are not satisfied—this is understandable since our identifiability condition is sufficient but not necessary. Figure 4 depicts the NMSE performance when different numbers of anchor nodes are available. In all three figures, there is a slowly decreasing trend in edge estimation MSE with L increasing, since wider window lengths yield improved estimates of the correlation matrices per window. Meanwhile, it can be seen that collecting a moderate number of time windows could already lead to reliable estimation, and the performance does not change much as the number keeps increasing (see $M=15$ vs $M=20$). Figure 4 also reveals a much better result when more anchor nodes are present, which better helps resolve the permutation and scaling ambiguities. It can also be readily observed that reliable performance can be obtained with only 5 anchor nodes.

We further compare the proposed joint diagonalization method with the CPSEM in [27] in terms of NMSE. Recall

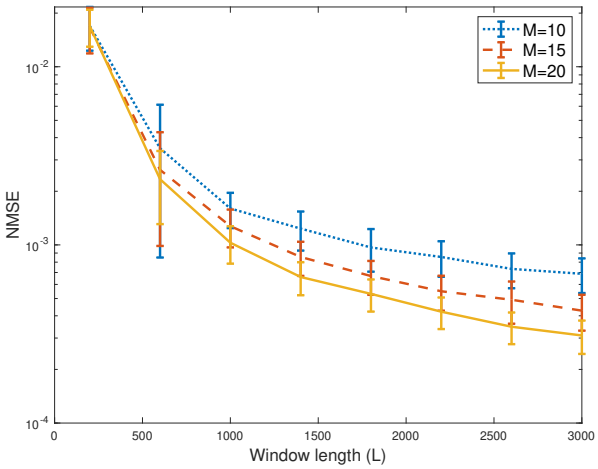


Fig. 2: NMSE of JDSEM for different window lengths, with $N_a=5$ anchor nodes.

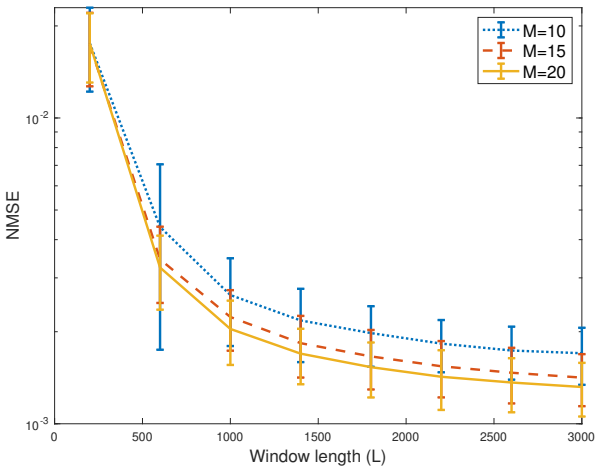


Fig. 3: NMSE of JDSEM for different window lengths, with $N_a=5$ anchor nodes, when the graph has a single-edge node.

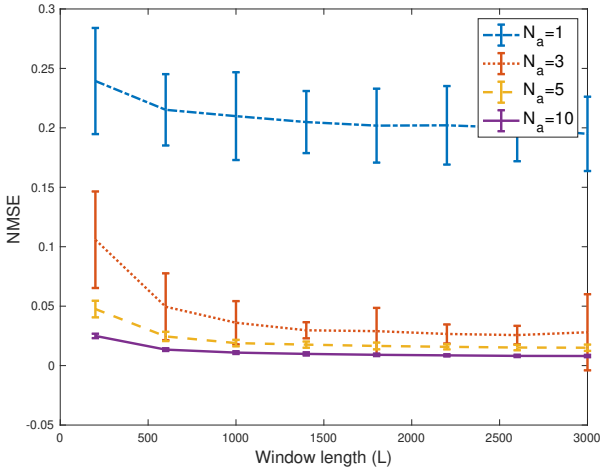


Fig. 4: NMSE plot of JDSEM for different window lengths and numbers of anchor nodes, with $M = 5$ windows.

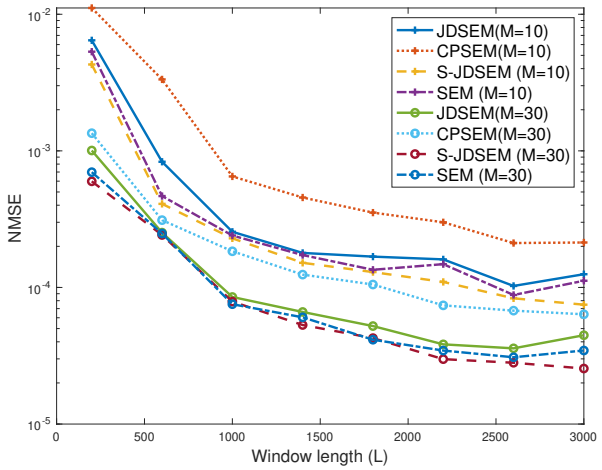


Fig. 6: Comparison of recovery NMSE for JDSEM, S-JDSEM, CPSEM and SEM versus window lengths, with $N_a = 3$.

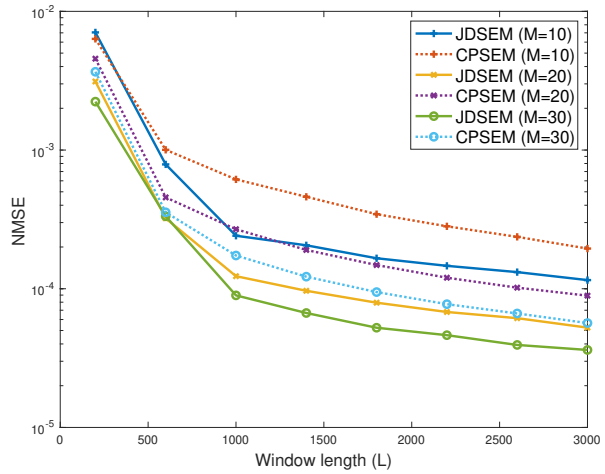


Fig. 5: Comparison of NMSE for JDSEM with $N_a = 10$, and CPSEM with fully observed exogenous inputs versus different window lengths.

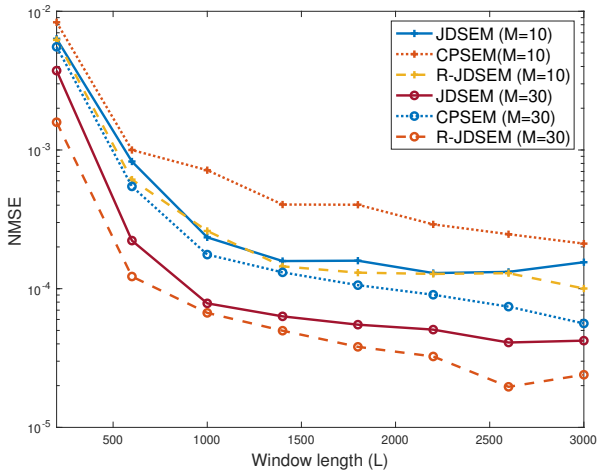


Fig. 7: NMSE of R-JDSEM and CPSEM [28] for different window lengths, with $N_a = 10$ anchor nodes, $\sigma_e^2 = 1$.

that instead of introducing anchor nodes, CPSEM assumes availability of $\mathbf{R}^x$, and for this reason we will assume full knowledge of $\mathbf{R}^x$ for CPSEM. One can observe from Figure 5 that CPSEM does not provide as good NMSE performance as the JDSEM.

B. Topology identification with S-JDSEM

This subsection aims at testing the performance of sparse JDSEM when the network structure is sparse structure, that is, the number of neighbors is much smaller than the total number of nodes.

Data generation. A random unweighted directed network of size 64 was generated with 6 outgoing links per node. The weighted adjacency matrix $\mathbf{A}$ was then constructed with weights drawn i.i.d. from the normal distribution $\mathcal{N}(0, 1)$, and the exogenous inputs $\mathbf{x}_t$ and $\mathbf{y}_t$ were generated as in the previous experiment.

Test results. Figure 6 illustrates the performance of S-JDSEM, JDSEM and CPSEM in terms of recovery NMSE. It can be readily observed that the novel JDSEM algorithms can recover the network topology with better accuracy than CPSEM, and the sparsity-aware S-JDSEM achieves the best performance because it exploits sparsity in the network structure, and also avoids matrix inversion. In addition, we compared our joint-diagonalization based method with the conventional SEM, where exact measurements of the nodal process are available, and an ℓ_1-norm regularizer is introduced to promote the sparse network connectivity, see e.g., [2], [4]. It can been seen that using only statistical information, the S-JDSEM can afford comparable recovery performance as SEM which has access to the exact nodal measurements. This is possible due to the existence of anchor nodes, and the fact that the model exploits the information present in the second-order statistics.

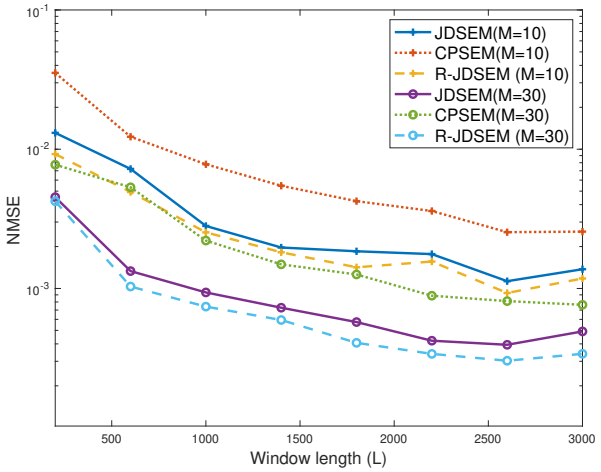


Fig. 8: NMSE of R-JDSEM and CPSEM [28] for different window lengths, with $N_a = 10$ anchor nodes, and $\sigma_e^2 = 5$.

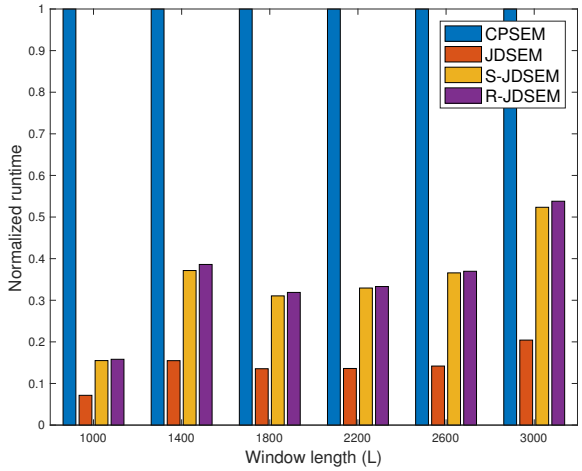


Fig. 9: Runtime performance with synthetic data.

C. Topology identification with R-JDSEM

In the present experiment, the performance of R-JDSEM is tested in the presence of outlying nodes.

Data generation. A random network with $N = 64$ nodes was generated from a prescribed "seed matrix," and endogenous variables $\{\mathbf{y}_t\}$ were generated using the SEM; that is, $\mathbf{y}_t = (\mathbf{I}-\mathbf{A})^{-1}(\mathbf{B}\mathbf{x}_t+\mathbf{e}_t)$, where $\mathbf{B}$ is an identity matrix. Vector $\mathbf{e}_t$ is sparse with 5% of nonzero entries signifying outlier nodes, and with values of the nonzero entries drawn from a Gaussian distribution $\mathcal{N}(0, \sigma_e^2)$ having $\sigma_e^2 = 1$ for Figure 7, and $\sigma_e^2 = 5$ for Figure 8; hence, they are comparable with the signal, and can thus be considered to be sparse outliers.

Test results. Figures 7 and 8 compare the performance of CPSEM, R-JDSEM, and JDSEM. Evidently, R-JDSEM is the most reliable one, while the JD-based algorithms outperform the CPSEM. This confirms that R-JDSEM is capable of taking into account the presence of outliers. All in all, we have demonstrated that the proposed JDSEM approaches can indeed recover the true network topologies with reliable performance even when exogenous variables are completely unknown.

In addition, Figure 9 illustrates the runtime comparison competing alternatives. It can be observed that besides being most efficient, the novel JD-based algorithms are more scalable than the CPSEM in [27].

D. Tests on real stock dataset

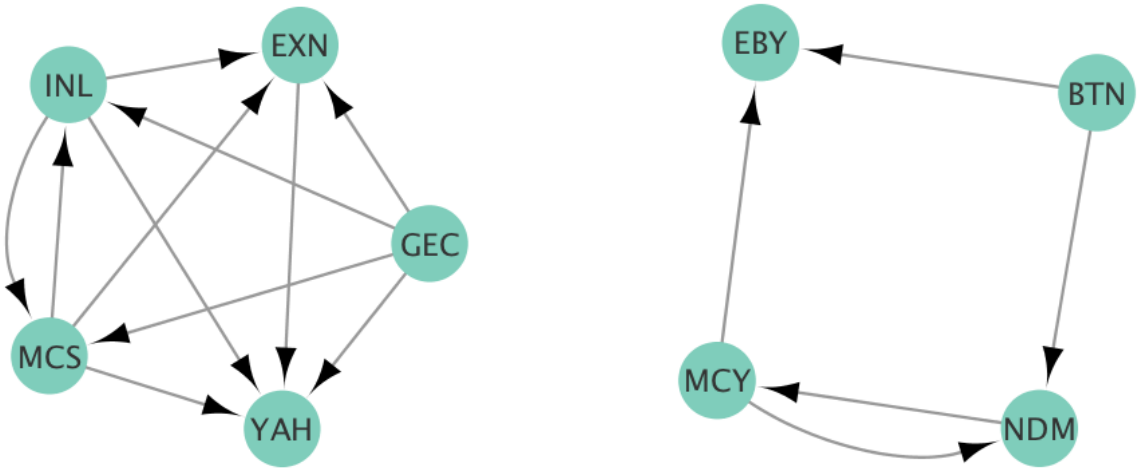
In this section, the performance of the proposed and algorithms is tested on real financial networks.

Dataset description. To conduct tests on real-world networks, we followed the procedure in [28], where historical stock price data were downloaded through a free *Yahoo* application program interface (API). Historical closing prices were obtained as time series for dates ranging from December 23, 2011 to September 30, 2016 ($1,200$ days in total). The stock time series from two groups of stocks were used: a) large technology companies (*Exxon-Mobil, Intel, Microsoft, Yahoo, and General Electric*), and b) online and brick-and-mortar retailers (*Bon-Ton, E-bay, Macy's, and Nordstrom*).

Test results. For this set of experiments, the combined multivariate time series was adopted as endogenous variables $\left(\{\mathbf{y}_t\}_{t=1}^{1,200}\right)$, after a pre-processing step in which samples were centered to have zero mean. Furthermore, money invested in the stocks constitutes exogenous inputs $\left(\{\mathbf{x}_t\}_{t=1}^{1,200}\right)$, which are not known in this case, since such information is generally not public, hence $\Omega = \emptyset$. Furthermore, it was observed that most stock prices tend to exhibit steady quarterly trends (rising or falling), and the window length was consequently set to $L \in \{80, 100, 120, 150, 200\}$. The CPSEM method proposed in [28] was run with $\Omega = \emptyset$ to infer causal dependencies between the selected stock prices. For the proposed JDSEM algorithms, the anchor nodes were selected uniformly at random in each experiment, of which the in-links were assumed known (obtained from the CPSEM output).

We conducted 100 independent runs of CPSEM with random initializations, while we randomly selected N_a anchor nodes for S-JDSEM, and R-JDSEM. It turned out that most estimates yielded identical support for $\hat{\mathbf{A}}$, with very slight variations in actual values of the entries. The most frequent network topologies from 100 independent experiments were deemed as the inferred network topologies. All algorithms reached the same estimated topologies as reported in [28]; see Figure 10. The figure shows strong dependencies in the group of technology companies, while the second plot shows stronger inter-dependencies between Macy's and Nordstrom than the others. Note that both Macy's and Nordstrom are well-known "brick-and-mortar" retailers and competitors. The stronger dependence between them seems to agree with the expectation that changes in the price of one would be expected to indirectly impact the other, which is also consistent with the result in [28].

The corresponding runtime of the proposed and algorithms is depicted in Figure 11 and 12. Evidently, JDSEM and S-JDSEM are much more scalable than CPSEM, while JDSEM and S-JDSEM need less than 1% of that of CPSEM, which again corroborates the effectiveness of the proposed method. Note that the runtime comparison in Figures 11 and 12 is



(a) (b)

Fig. 10: Visualization of network topologies inferred from the stock price time series, depicting: a) technology companies; and b) online and "brick-and-mortar" retailers. Notice the stronger dependencies between the two competing "brick-and-mortar" retailers, Macy's (MCY) and Nordstrom (NDM) [28].

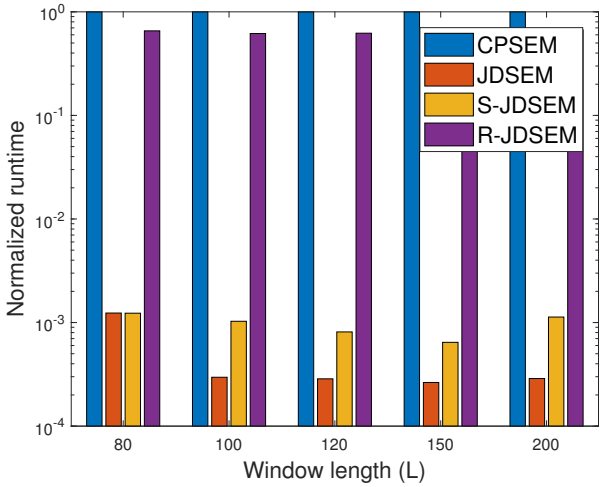


Fig. 11: Runtime for technology companies involving different window lengths (time segments), and $N_a = 2$ anchor nodes.

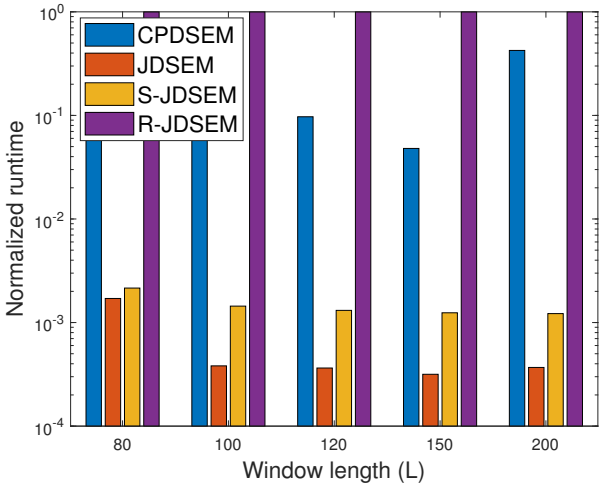


Fig. 12: Runtime for retailer companies involving different window lengths, and $N_a = 2$ anchor nodes.

clearly different from Figure 9, especially for R-JDSEM. The main reason for the difference in the run time is that the algorithms involve iterative procedures, and the convergence may largely depend on the property of data. The longer runtime of R-JDSEM in Fig 9 and 10 is likely due to the fact that the real data does not include significant outliers, which leads to a slower convergence rate of the R-JDSEM.

We further tested the performance of R-JDSEM in the presence of outliers, where we randomly selected an outlier for the retailer network, and treated the result of the CPSEM using true data as ground truth. Figure 13 shows the NMSE of JDSEM and R-JDSEM, corroborating that the R-JDSEM is indeed more robust than JDSEM in presence of outliers.

VI. Conclusions

This paper put forth a novel framework for inferring network topologies from statistics of nodal processes. The task was formulated as a joint diagonalization problem. Novel algorithms were developed and shown capable of leveraging prior knowledge to account for edge sparsity and gain resilience to outliers. Numerical tests on both synthetic and real data corroborated the effectiveness of the proposed approaches.

To broaden the scope of this study, there are several intriguing directions to pursue: a) more scalable algorithms for large-scale networks; b) online real-time algorithms for time-varying networks; c) distributed implementations that are well-motivated for large-scale networks, and d) nonlinear tensor-based network topology inference.

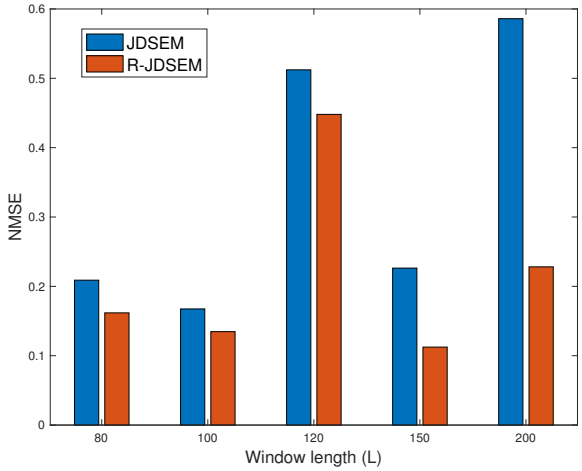


Fig. 13: NMSE for retailer companies involving different window lengths in presence of outlier.

APPENDIX

A. ADMM solver for (28)

By applying ADMM, the non-smooth part of the objective function, which is induced by the ℓ_1 norm can be decoupled from the smooth part. For each subproblem in (23), introducing the auxiliary variable $\mathbf{c}_n$ leads to

$$\begin{aligned}&\min_{\bar{\mathbf{h}}_n} \|\mathbf{d}_n\|_1 + \lambda\|\mathbf{c}_n\|_1\\ &\text{s.t.}\quad \bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n = \mathbf{d}_n,\quad \bar{\mathbf{h}}_n = \mathbf{c}_n\ .\end{aligned} \tag{37}$$

Hence, the augmented Lagrangian of (37) can be written as

$$\begin{aligned}&\mathcal{L}_n(\bar{\mathbf{h}}_n, \mathbf{c}_n, \mathbf{u}_n, \mathbf{d}_n, \mathbf{v}_n) = \|\mathbf{d}_n\|_1 + \lambda\|\mathbf{c}_n\|_1\\ &+\mathbf{v}_n^\top(\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n - \mathbf{d}_n) + \mathbf{u}_n^\top(\bar{\mathbf{h}}_n - \mathbf{c}_n)\\ &+\frac{\rho}{2}\|\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n - \mathbf{d}_n\|_2^2 + \frac{\rho}{2}\|\bar{\mathbf{h}}_n - \mathbf{c}_n\|_2^2\end{aligned} \tag{38}$$

where $\mathbf{u}_n$ denotes the Lagrange multiplier, while ρ is a positive scalar. ADMM essentially adopts alternating minimization iterations over the variables $\bar{\mathbf{h}}_n$ and $\mathbf{c}_n$, followed by a gradient ascent step over the multiplier $\mathbf{u}_n$ [7], [25]. During the $(k+1)$st iteration, updates of the variables are given by

$$\bar{\mathbf{h}}_n^{k+1} = \arg\min_{\bar{\mathbf{h}}_n} \mathcal{L}_n(\bar{\mathbf{h}}_n, \mathbf{c}_n^k, \mathbf{u}_n^k, \mathbf{d}_n^k, \mathbf{v}_n^k) \tag{39a}$$

$$\mathbf{c}_n^{k+1} = \arg\min_{\mathbf{c}_n} \mathcal{L}_n(\bar{\mathbf{h}}_n^{k+1}, \mathbf{c}_n, \mathbf{u}_n^k, \mathbf{d}_n^k, \mathbf{v}_n^k) \tag{39b}$$

$$\mathbf{d}_n^{k+1} = \arg\min_{\mathbf{c}_n} \mathcal{L}_n(\bar{\mathbf{h}}_n^{k+1}, \mathbf{c}_n^{k+1}, \mathbf{u}_n^k, \mathbf{d}_n, \mathbf{v}_n^k) \tag{39c}$$

$$\mathbf{v}_n^{k+1} = \mathbf{v}_n^k + \rho(\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n^{k+1} + \boldsymbol{\epsilon}_n - \mathbf{d}_n^{k+1}) \tag{39d}$$

$$\mathbf{u}_n^{k+1} = \mathbf{u}_n^k + \rho(\bar{\mathbf{h}}_n^{k+1} - \mathbf{c}_n^{k+1}). \tag{39e}$$

Per step, the augmented Lagrangian is minimized w.r.t. a specific variable, with all the rest staying fixed to their most recent updates, until convergence is attained.

Algorithm 2 ADMM solver for (28)

Input: $a_{nj},\ j \in \Omega_n$, $\{\mathbf{R}_m^y\}$, current estimates of $\{\mathbf{h}_i\}_{i=1}^N$
Initialization: $\bar{h}_n^1 = \mathbf{0}$, $\mathbf{c}_n^1 = \mathbf{0}$, $\mathbf{u}_n^1 = \mathbf{0}$
Construct $\mathbf{W}_n$ and $\boldsymbol{\epsilon}_n$ via (17) and (19).
for $k = 1, \ldots$ **do**
$\bar{\mathbf{h}}_n^{k+1} = \big(\bar{\mathbf{W}}_n^\top\bar{\mathbf{W}}_n + \rho\mathbf{I}\big)^{-1}(\rho\mathbf{c}_n^k - \mathbf{u}_n^i unk - \bar{\mathbf{W}}_n^\top\boldsymbol{\epsilon}_n)$
$\mathbf{c}_n^{k+1} = \mathcal{P}_{\lambda/\rho}(\bar{\mathbf{h}}_n^{k+1} + \mathbf{u}_n^k/\rho)$
$\mathbf{d}_n^{k+1} = \mathcal{P}_{1/\rho}(\bar{\mathbf{h}}_n^{k+1} + \bar{\mathbf{W}}_n\bar{\mathbf{h}}_n + \boldsymbol{\epsilon}_n + \mathbf{v}_n^k/\rho)$
$\mathbf{u}_n^{k+1} = \mathbf{u}_n^k + \rho(\bar{\mathbf{h}}_n^{k+1} - \mathbf{c}_n^{k+1})$
$\mathbf{v}_n^{k+1} = \mathbf{v}_n^k + \rho(\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n^{k+1} + \boldsymbol{\epsilon}_n - \mathbf{d}_n^{k+1})$
end for
$[\mathbf{h}_n]_{j\notin\Omega_n} = \bar{\mathbf{h}}_n^k$, $h_{nj} = -a_{nj},\ j \in \Omega_n$, $h_{nn} = 1$.
Output: $\mathbf{h}_n$

Focusing on (39a) and differentiating w.r.t. $\bar{\mathbf{h}}_n^{k+1}$ yields

$$\begin{aligned}&\big(\bar{\mathbf{W}}_n^\top\bar{\mathbf{W}}_n + \rho\mathbf{I}\big)\bar{\mathbf{h}}_n = \rho\mathbf{c}_n^k - \mathbf{u}_n^k - \bar{\mathbf{W}}_n^\top\boldsymbol{\epsilon}_n + \bar{\mathbf{W}}_n^\top\mathbf{d}_n\\ &\Rightarrow \bar{\mathbf{h}}_n^{k+1} = \big(\bar{\mathbf{W}}_n^\top\bar{\mathbf{W}}_n + \rho\mathbf{I}\big)^{-1}(\rho\mathbf{c}_n^k - \mathbf{u}_n^k - \bar{\mathbf{W}}_n^\top\boldsymbol{\epsilon}_n + \bar{\mathbf{W}}_n^\top\mathbf{d}_n).\end{aligned} \tag{40}$$

Next, the update of $\mathbf{c}_n^{k+1}$ in (39b) can be cast as

$$\mathbf{c}_n^{k+1} = \arg\min_{\mathbf{c}_n} \lambda\|\mathbf{c}_n\|_1 + \frac{\rho}{2}\|\mathbf{c}_n - \bar{\mathbf{h}}_n^{k+1} - \mathbf{u}_n^k/\rho\|_2^2 \tag{41}$$

which admits the following closed-form solution

$$\mathbf{c}_n^{k+1} = \mathcal{P}_{\lambda/\rho}(\bar{\mathbf{h}}_n^{k+1} + \mathbf{u}_n^k/\rho) \tag{42}$$

where $\mathcal{P}_z(.)$ denotes the entrywise soft thresholding operator defined as

$$\mathcal{P}_\lambda(z) := \frac{z}{|z|}\max(|z| - \lambda, 0). \tag{43}$$

Similarly, $\mathbf{d}_n$ can be updated via

$$\mathbf{d}_n^{k+1} = \arg\min_{\mathbf{d}_n} \|\mathbf{d}_n\|_1 + \frac{\rho}{2}\|\mathbf{d}_n - \bar{\mathbf{W}}_n\bar{\mathbf{h}}_n^{k+1} - \boldsymbol{\epsilon}_n - \mathbf{v}_n^k/\rho\|_2^2 \tag{44}$$

which can again be solved in closed form as

$$\mathbf{d}_n^{k+1} = \mathcal{P}_{1/\rho}(\bar{\mathbf{W}}_n\bar{\mathbf{h}}_n^{k+1} + \boldsymbol{\epsilon}_n + \mathbf{v}_n^k/\rho)\ . \tag{45}$$

Together with the gradient ascent step over the dual variables in (39e) and (39d), Algorithm 2 summarizes the iterations resulting from the developed ADMM solver for (28).

REFERENCES

[1] B. Afsari, "Sensitivity analysis for the problem of matrix joint diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1148–1171, Sep. 2008.
[2] B. Baingana, G. Mateos, and G. Giannakis, "Dynamic structural equation models for tracking topologies of social networks," in *Proc. of Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Saint Martin, Dec. 2013, pp. 292–295.
[3] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Sig. Proc.*, vol. 8, no. 4, pp. 563–575, Aug. 2014.
[4] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Sig. Proc.*, vol. 58, no. 3, pp. 1847–1862, 2010.
[5] J. A. Bazerque, B. Baingana, and G. B. Giannakis, "Identifiability of sparse structural equation models for directed and cyclic networks," in *Proc. of Global Conf. on Signal and Info. Processing*, Austin, TX, Dec. 2013.

[6] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jul. 2011.

[8] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Comp. Biol.*, vol. 9, no. 5, p. e1003068, May 2013.

[9] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[10] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Dec. 2008.

[12] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. of the IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[13] A. S. Goldberger, "Structural equation methods in the social sciences," *Econometrica*, vol. 40, no. 6, pp. 979–1001, Nov. 1972.

[14] D. Kaplan, *Structural Equation Modeling: Foundations and Extensions*. Sage, 2009.

[15] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.

[16] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[17] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95–138, 1977.

[18] D. Le Bihan, J.-F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat, "Diffusion tensor imaging: concepts and applications," *Journal of Magnetic Resonance Imaging*, vol. 13, no. 4, pp. 534–546, 2001.

[19] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, Mar. 2010.

[20] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Transactions on Signal Processing*, vol. 63, no. 10, pp. 2663–2677, May 2015.

[21] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, Sep 2019.

[22] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 530–541, 2001.

[23] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[24] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proc. Intl. Conf. Mach. Learn.*, Bellevue, WA, USA, Jul. 2011.

[25] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links —- Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.

[26] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, May 2017.

[27] Y. Shen, B. Baingana, and G. B. Giannakis, "Inferring directed network topologies via tensor factorization," in *Proc. of Asilomar Conf.*, Pacific Grove, CA, Nov. 2016.

[28] ——, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Signal Processing*, vol. 65, no. 14, pp. 3675–3687, July 2017.

[29] Y. Shen, X. Fu, G. B. Giannakis, and N. D. Sidiropoulos, "Directed network topology inference via sparse joint diagonalization," in *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2017, pp. 698–702.

[30] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, July 2017.

[31] A. Stegeman and N. D. Sidiropoulos, "On Kruskal's uniqueness condition for the CAMDECOMP/PAEAFAC decomposition," *Linear Algebra and its Applications*, vol. 420, no. 2, pp. 540–552, Jan. 2007.

[32] P. Tichavsky and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878–891, 2008.

[33] R. Vollgraf and K. Obermayer, "Quadratic optimization for simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3270–3278, 2006.

[34] M. Wax and J. Sheinvald, "A least-squares approach to joint diagonalization," *IEEE Signal Processing Letters*, vol. 4, no. 2, pp. 52–53, 1997.

[35] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[36] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on signal processing*, vol. 50, no. 7, pp. 1545–1553, 2002.

[37] A. Ziehe, P. Laskov, G. Nolte, and K.-R. MÃžller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *Journal of Machine Learning Research*, vol. 5, no. Jul, pp. 777–800, 2004.