Dynamic Structural Equation Models for Tracking Topologies of Social Networks[†]

Brian Baingana, Gonzalo Mateos, and Georgios B. Giannakis
Dept. of ECE, University of Minnesota, Minneapolis, MN 55455

Abstract— **Many real-world processes evolve in cascades over complex networks, whose topologies are often unobservable and change over time. However, the so-termed adoption times when blogs mention popular news items, individuals in a community catch an infectious disease, or consumers adopt a trendy electronics product are typically known, and are implicitly dependent on the underlying network. To infer the network topology, a *dynamic* structural equation model is adopted that captures the relationship between observed adoption times and the unknown edge weights. Assuming a slowly time-varying network and leveraging the sparse connectivity inherent to social networks, edge weights are estimated by minimizing a sparsity-regularized exponentially-weighted least-squares criterion. An alternating-direction method of multipliers solver is developed to this end, and preliminary tests on synthetic network data corroborate the effectiveness of the novel algorithm in unveiling the dynamically-evolving network topology.**

I. Introduction

Networks arising in natural and man-made settings provide the backbone for the propagation of *contagions* such as the spread of popular news stories, the adoption of buying trends among consumers, and the spread of infectious diseases. For example, a terrorist attack may be reported within minutes on mainstream news websites. A cascade emerges because these websites' readership typically includes bloggers who write about the attack as well, influencing their own readers in turn to do the same. Although the times when "nodes" get infected are often observable, the underlying network topologies over which cascades propagate are typically unknown and dynamic. Knowledge of the topology plays a crucial role for several reasons e.g., when social media advertisers select a small set of initiators so that an online campaign can go viral, or when healthcare initiatives wish to infer hidden needle-sharing networks of injecting drug users.

Inference of networks using temporal traces of infection events has recently become an active area of research. Several prior approaches postulate probabilistic models and leverage maximum likelihood estimation (MLE) to infer edge weights as pairwise transmission rates between nodes [7], [6]. However, these methods assume that the network does not change over time. A dynamic algorithm has been recently proposed to infer time-varying diffusion networks by solving an MLE problem via stochastic gradient descent iterations [8]. Although successful experiments on large-scale web data reliably uncover information pathways, the estimator in [8] does not explicitly account for edge sparsity prevalent in social and information networks. Moreover, most prior approaches only attribute node infection events to the network topology, and do not account for the influence of external sources such as a ground crew for a mainstream media website.

The present paper proposes a *dynamic* structural equation model (SEM) for the network, which postulates that node infection times depend on both topological and external influences. Topological influences are modeled as linear combinations of infection times of other nodes in the network, whose weights correspond to entries in the time-varying adjacency matrix. Accounting for external influences is well motivated by drawing upon examples from online media, where established news websites depend more on on-site reporting than blog references. SEMs have recently been proposed for *static* gene regulatory network inference from gene expression data; see e.g., [3] and references therein. They are attractive because of their simplicity, and ability to capture edge directionalities.

Supposing the network varies slowly with time, parameters in the proposed dynamic SEM are estimated adaptively by minimizing a sparsity-promoting exponentially-weighted least-squares (LS) criterion. To account for the inherently sparse connectivity of social networks, an ℓ_1-norm regularization term that promotes sparsity on the entries of the network adjacency matrix is incorporated in the cost function; see also [4] and [1]. A novel algorithm to jointly track the network's adjacency matrix and the weights capturing the level of external influences is developed, which adaptively minimizes the resulting non-differentiable cost function via the alternating-direction method of multipliers (ADMoM); see e.g., [2]. The resulting algorithm is provably convergent, and offers closed-form updates per iteration. The remainder of the paper is organized as follows: Section II describes the dynamic SEM and states the network topology inference problem, while Section III develops the proposed topology-tracking algorithm. Section IV presents preliminary numerical tests, and Section V concludes the paper.

Notation. Bold uppercase (lowercase) letters will denote matrices (column vectors), while operators $(\cdot)^\top$ and $\text{Tr}(\cdot)$ will stand for matrix transposition and trace, respectively. The $N \times N$ identity matrix will be represented by $\mathbf{I}_N$; $\|.\|_p$, and $\|.\|_F$ will denote ℓ_p and Frobenius norms, respectively.

II. Network Model and Problem Statement

Consider a dynamic network with N nodes observed over time intervals $t = 1, \ldots, T$, whose topology is described by an unknown, time-varying, and weighted adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{N \times N}$. Entry (i, j) of $\mathbf{A}^t$ is nonzero only if an edge connects nodes i and j during the time interval t, as illustrated in the 8-node network in Fig. 1. Suppose C contagions propagate

[†] Work in this paper was supported by the NSF ECCS Grant No. 1202135 and the NSF AST Grant No. 1247885.

978-1-4673-3146-3/13/$31.00 ©2013IEEE

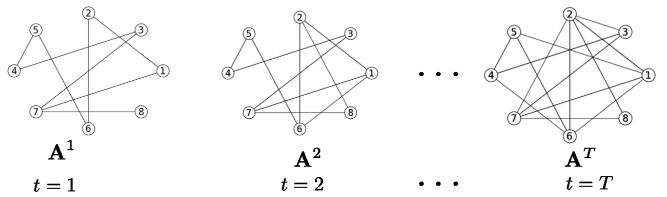

Fig. 1: Dynamic network observed across time intervals.

over the network, and the difference between infection time of node i by contagion c and the earliest observation time is denoted by y_{ic}^t. In online media, y_{ic}^t can be obtained by recording the time when website i mentions news item c. For uninfected nodes at slot t, y_{ic}^t is set to an arbitrarily large number. Assume that the susceptibility x_{ic} of node i to external (non-topological) infection by contagion c is known and time invariant over the observation interval. In the web context, x_{ic} can be set to the search engine rank of website i with respect to (w.r.t.) keywords associated with c.

Infection time of node i during interval t is modeled as

$$y_{ic}^t = \sum_{j \neq i} a_{ij}^t y_{jc}^t + b_{ii}^t x_{ic} + e_{ic}^t \tag{1}$$

where a_{ij}^t denotes entry (i,j) of $\mathbf{A}^t$, b_{ii}^t captures the time-varying level of influence of external sources, and e_{ic}^t accounts for measurement errors and unmodeled dynamics. Rewriting (1) for the entire network leads to the dynamic SEM

$$\mathbf{y}_c^t = \mathbf{A}^t \mathbf{y}_c^t + \mathbf{B}^t \mathbf{x}_c + \mathbf{e}_c^t \tag{2}$$

where the $N \times 1$ vector $\mathbf{y}_c^t := [y_{1c}^t, \ldots, y_{Nc}^t]^\top$ collects the node infection times by contagion c during interval t, and $\mathbf{B}^t := \text{diag}(b_{11}^t, \ldots, b_{NN}^t)$. Similarly, $\mathbf{x}_c := [x_{1c}, \ldots, x_{Nc}]^\top$ and $\mathbf{e}_c^t := [e_{1c}^t, \ldots, e_{Nc}^t]^\top$. Collecting observations for all C contagions yields the dynamic matrix SEM

$$\mathbf{Y}^t = \mathbf{A}^t \mathbf{Y}^t + \mathbf{B}^t \mathbf{X} + \mathbf{E}^t \tag{3}$$

where $\mathbf{Y}^t := [\mathbf{y}_1^t, \ldots, \mathbf{y}_C^t]$, $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_C]$, and $\mathbf{E}^t := [\mathbf{e}_1^t, \ldots, \mathbf{e}_C^t]$ are all $N \times C$ matrices.

Given $\{\mathbf{Y}^t\}_{t=1}^T$ and $\mathbf{X}$, the goal is to track the underlying network topology $\{\mathbf{A}^t\}_{t=1}^T$ and the effect of external influences $\{\mathbf{B}^t\}_{t=1}^T$. To this end, the novel algorithm developed in the next section assumes slow time variation of the network topology and leverages the inherent sparsity of edges that is typical of social networks.

III. Topology Tracking Algorithm

This section deals with a regularized LS approach to estimating $\{\mathbf{A}^t, \mathbf{B}^t\}$ in (3). In a *static* setting with all measurements $\{\mathbf{Y}^t\}_{t=1}^T$ available, one solves the batch problem

$$\begin{aligned} \{\hat{\mathbf{A}}, \hat{\mathbf{B}}\} = \arg\min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2} \sum_{t=1}^{T} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 + \lambda \|\mathbf{A}\|_1 \\ \text{s. to} \quad & a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j \end{aligned} \tag{4}$$

where $\|\mathbf{A}\|_1 := \sum_{i,j} |a_{ij}|$ is a sparsity-promoting regularization, and $\lambda > 0$ controls the sparsity level of $\hat{\mathbf{A}}$. Absence of a self-loop at node i is enforced by the constraint $a_{ii} = 0$.

A. Exponentially-weighted LS estimator

In practice, measurements are typically acquired in a sequential manner and the sheer scale of social networks calls for estimation algorithms with minimal storage requirements. Recursive solvers enabling sequential inference of the underlying network topology are thus preferred. Moreover, introducing a "forgetting factor" that assigns more weight to the most recent residuals makes it possible to track temporal variations of the topology. For $\tau = 1, \ldots, T$, the sparsity-regularized exponentially-weighted LS estimator

$$\begin{aligned} \{\hat{\mathbf{A}}^\tau, \hat{\mathbf{B}}^\tau\} = \arg\min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2} \sum_{t=1}^{\tau} \beta^{\tau-t} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 \\ & + \lambda_\tau \|\mathbf{A}\|_1 \\ \text{s. to} \quad & a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j \end{aligned} \tag{5}$$

where $\beta \in (0,1]$ and $\beta^{\tau-t}$ is the forgetting factor, which forms estimates $\{\hat{\mathbf{A}}^\tau, \hat{\mathbf{B}}^\tau\}$ using measurements acquired until time τ. Notice that λ_τ is allowed to vary with time in order to capture the generally changing edge sparsity.

B. ADMoM solver

Exploiting the problem structure in (5), an alternating-direction method of multipliers (ADMoM) algorithm is developed to track the network topology. Leaving the equality constraints ($a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j$) temporarily implicit and introducing a dummy variable $\mathbf{C} \in \mathbb{R}^{N \times N}$, leads to the following equality constrained optimization problem:

$$\begin{aligned} \{\hat{\mathbf{A}}^\tau, \hat{\mathbf{B}}^\tau, \hat{\mathbf{C}}^\tau\} = \arg\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \quad & \frac{1}{2} \sum_{t=1}^{\tau} \beta^{\tau-t} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 \\ & + \lambda_\tau \|\mathbf{C}\|_1 \\ \text{s. to} \quad & \mathbf{A} = \mathbf{C}. \end{aligned} \tag{6}$$

With $\rho > 0$ denoting the penalty parameter and $\mathbf{\Gamma}$ the matrix of dual variables, the augmented Lagrangian for (6) is

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Gamma}) \quad := \quad & \frac{1}{2} \sum_{t=1}^{\tau} \beta^{\tau-t} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 \\ & + \lambda_\tau \|\mathbf{C}\|_1 + \text{Tr}\{\mathbf{\Gamma}^\top (\mathbf{A} - \mathbf{C})\} \\ & + \frac{\rho}{2} \|\mathbf{A} - \mathbf{C}\|_F^2. \end{aligned} \tag{7}$$

During iteration $r+1$ of the ADMoM algorithm, alternating minimization of $\mathcal{L}_\rho(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Gamma})$ w.r.t. $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, followed by an update of the dual variables yields

$$\begin{aligned} \mathbf{A}_{r+1} = \quad & \arg\min_{\mathbf{A}} \ \tfrac{1}{2} \sum_{t=1}^{\tau} \beta^{\tau-t} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}_r\mathbf{X}\|_F^2 \\ & + \tfrac{\rho}{2} \|\mathbf{A} - \mathbf{C}_r\|_F^2 + \text{Tr}\{\mathbf{\Gamma}_r^\top (\mathbf{A} - \mathbf{C}_r)\} && \text{(8a)} \\ \mathbf{B}_{r+1} = \quad & \arg\min_{\mathbf{B}} \ \tfrac{1}{2} \sum_{t=1}^{\tau} \beta^{\tau-t} \|\mathbf{Y}^t - \mathbf{A}_{r+1}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 && \text{(8b)} \\ \mathbf{C}_{r+1} = \quad & \arg\min_{\mathbf{C}} \ \tfrac{\rho}{2} \|\mathbf{A}_{r+1} - \mathbf{C}\|_F^2 + \text{Tr}\{\mathbf{\Gamma}_r^\top (\mathbf{A}_{r+1} - \mathbf{C})\} \\ & + \lambda_\tau \|\mathbf{C}\|_1 && \text{(8c)} \\ \mathbf{\Gamma}_{r+1} = \quad & \mathbf{\Gamma}_r + \rho \left(\mathbf{A}_{r+1} - \mathbf{C}_{r+1}\right). && \text{(8d)} \end{aligned}$$

As shown next, subproblems (8a)-(8c) can be solved in closed form and the resulting algorithm provably converges to global optima for each τ [2]. Turning attention to solving for $\mathbf{A}_{r+1}$, note that (8a) can be recast as

$$\begin{aligned}\mathbf{A}_{r+1} \;=\; \arg\min_{\mathbf{a}_1,\ldots,\mathbf{a}_N} \sum_{i=1}^{N} \Bigg[& \frac{1}{2}\mathbf{a}_i^\top(\mathbf{P}^\tau+\rho\mathbf{I}_N)\mathbf{a}_i \\ & -\mathbf{a}_i^\top(\rho\mathbf{c}_{i,r}+\mathbf{p}_i^\tau-\boldsymbol{\gamma}_{i,r}-b_{ii,r}\mathbf{Q}^\tau\mathbf{x}_i)\Bigg] \quad (9)\end{aligned}$$

where constant terms w.r.t. $\mathbf{A}$ have been dropped and $\mathbf{a}_i^\top$, $\mathbf{c}_{i,r}^\top$, and $\boldsymbol{\gamma}_{i,r}^\top$ correspond to row i of $\mathbf{A}$, $\mathbf{C}_r$, and $\boldsymbol{\Gamma}_r$ respectively. In addition, $\mathbf{P}^\tau := \sum_{t=1}^{\tau}\beta^{\tau-t}\mathbf{Y}^t(\mathbf{Y}^t)^\top$, $\mathbf{p}_i^\tau := \sum_{t=1}^{\tau}\beta^{\tau-t}\mathbf{Y}^t\mathbf{y}_i^t$ with $(\mathbf{y}_i^t)^\top$ corresponding to row i of $\mathbf{Y}^t$, and $\mathbf{Q}^\tau := \sum_{t=1}^{\tau}\beta^{\tau-t}\mathbf{Y}^t$. Interestingly, the data-related quantities $\mathbf{P}^\tau$, $\mathbf{p}_i^\tau$, and $\mathbf{Q}^\tau$ can be recursively updated as follows:

$$\begin{aligned}\mathbf{P}^\tau &= \beta\mathbf{P}^{\tau-1}+\mathbf{Y}^\tau(\mathbf{Y}^\tau)^\top && (10a)\\ \mathbf{p}_i^\tau &= \beta\mathbf{p}_i^{\tau-1}+\mathbf{Y}^\tau\mathbf{y}_i^\tau && (10b)\\ \mathbf{Q}^\tau &= \beta\mathbf{Q}^{\tau-1}+\mathbf{Y}^\tau. && (10c)\end{aligned}$$

The quadratic cost in (9) decouples across rows of $\mathbf{A}$, and can be efficiently solved per row in parallel and in closed form. For row i, the constraint $a_{ii}=0$ is incorporated by solving

$$\tilde{\mathbf{a}}_{i,r+1} = \arg\min_{\tilde{\mathbf{a}}_i}\frac{1}{2}\tilde{\mathbf{a}}_i^\top(\mathbf{P}_i^\tau+\rho\mathbf{I}_{N-1})\tilde{\mathbf{a}}_i-\tilde{\mathbf{a}}_i^\top\mathbf{w}_i^\tau \qquad (11)$$

where $\tilde{\mathbf{a}}_i$ denotes the $(N-1)\times 1$ vector obtained by removing entry i from $\mathbf{a}_i$. Similarly, $\mathbf{P}_i^\tau$ is obtained by removing row i and column i from $\mathbf{P}^\tau$, and $\mathbf{w}_i^\tau$ is obtained by removing entry i from $\rho\mathbf{c}_{i,r}+\mathbf{p}_i^\tau-\boldsymbol{\gamma}_{i,r}-b_{ii,r}\mathbf{Q}^\tau\mathbf{x}_i$. Solving (11) yields

$$\tilde{\mathbf{a}}_{i,r+1} = (\mathbf{P}_i^\tau+\rho\mathbf{I}_{N-1})^{-1}\mathbf{w}_i^\tau \qquad (12)$$

and row i of $\mathbf{A}_{r+1}$ is updated by setting

$$\begin{aligned}\mathbf{a}_{i,r+1}^\top \;=\; & [\tilde{a}_{i1,r+1},\ldots,\tilde{a}_{ii-1,r+1},0,\\ & \tilde{a}_{ii,r+1},\ldots,\tilde{a}_{iN-1,r+1}]. \quad (13)\end{aligned}$$

Next, setting $b_{ij}=0$ for all off-diagonal entries of $\mathbf{B}$, it turns out that (8b) amounts to solving N scalar problems

$$\begin{aligned}\arg\min_{b_{11},\ldots,b_{NN}} \sum_{i=1}^{N}\sum_{t=1}^{\tau}\beta^{\tau-t}\Bigg[& \frac{1}{2}b_{ii}^2\mathbf{x}_i^\top\mathbf{x}_i - b_{ii}\mathbf{y}_i^{t\top}\mathbf{x}_i \\ & +b_{ii}\mathbf{a}_{i,r+1}^\top\mathbf{Y}^t\mathbf{x}_i\Bigg] \quad (14)\end{aligned}$$

which yields the per-entry closed-form solution

$$b_{ii,r+1} = \frac{(\mathbf{q}_i^\tau)^\top\mathbf{x}_i-\mathbf{a}_{i,r+1}^\top\mathbf{Q}^\tau\mathbf{x}_i}{\mu^\tau\mathbf{x}_i^\top\mathbf{x}_i} \qquad (15)$$

where $(\mathbf{q}_i^\tau)^\top$ denotes row i of $\mathbf{Q}^\tau$, and $\mu^\tau := (1-\beta^\tau)/(1-\beta)$.

To solve (8c) it is prudent to rewrite the cost function in terms of the rows of matrices $\mathbf{A}_{r+1}$, $\boldsymbol{\Gamma}_r$ and $\mathbf{C}$, leading to

$$\arg\min_{\mathbf{c}_1,\ldots,\mathbf{c}_N}\sum_{i=1}^{N}\left[\frac{\rho}{2}\|\mathbf{a}_{i,r+1}-\mathbf{c}_i\|_2^2-\mathbf{c}_i^\top\boldsymbol{\gamma}_{i,r}+\lambda_\tau\|\mathbf{c}_i\|_1\right]. \quad (16)$$

Upon defining $\boldsymbol{\alpha}_{i,r+1} := \mathbf{a}_{i,r+1}+\frac{1}{\rho}\boldsymbol{\gamma}_{i,r}$ and completing the square in (16), the update step for matrix $\mathbf{C}$ can be cast as

$$\arg\min_{\mathbf{c}_1,\ldots,\mathbf{c}_N}\sum_{i=1}^{N}\left[\frac{1}{2}\|\boldsymbol{\alpha}_{i,r+1}-\mathbf{c}_i\|_2^2+\frac{\lambda_\tau}{\rho}\|\mathbf{c}_i\|_1\right] \qquad (17)$$

Algorithm 1 ADMoM solver for topology tracking

```
1:  Input: ${\mathbf{Y}^\tau}_{\tau=1}^T$, $\mathbf{X}$, $\epsilon$, $\beta$, $\rho$
2:  Initialize $\mathbf{P}^0$, $\mathbf{Q}^0$, $\mathbf{B}^0$, $\boldsymbol{\Gamma}^0$, $\mathbf{C}^0$, $\mu^0$, $\lambda_0$
3:  for $\tau = 1,\ldots,T$ do
4:    Initialize $r = 0$
5:    Update $\mathbf{P}^\tau$, $\mathbf{Q}^\tau$, $\lambda_\tau$, $\mu^\tau$
6:    repeat
7:      for $i = 1\ldots N$ do
8:        Compute $\mathbf{w}_i^\tau$, $\mathbf{P}_i^\tau$, $\mathbf{p}_i^\tau$
9:        Compute $\tilde{\mathbf{a}}_{i,r+1} = (\mathbf{P}_i^\tau+\rho\mathbf{I}_{N-1})^{-1}\mathbf{w}_i^\tau$
10:       Update $\mathbf{a}_{i,r+1}$ via (13)
11:       Update $b_{ii,r+1}$ via (15)
12:       Update $\mathbf{c}_{i,r+1}$ via (19)
13:     end for
14:     $\boldsymbol{\Gamma}_{r+1} = \boldsymbol{\Gamma}_r + \rho(\mathbf{A}_{r+1}-\mathbf{C}_{r+1})$
15:     $r = r+1$
16:   until $\|\mathbf{A}_{r+1}-\mathbf{C}_{r+1}\|_F \le \epsilon$
17:   Return $\mathbf{A}^\tau = \mathbf{A}_r$, $\mathbf{B}^\tau = \mathbf{B}_r$
18: end for
```

which amounts to solving

$$\arg\min_{\mathbf{c}_i}\;\frac{1}{2}\|\boldsymbol{\alpha}_{i,r+1}-\mathbf{c}_i\|_2^2+\frac{\lambda_\tau}{\rho}\|\mathbf{c}_i\|_1 \qquad (18)$$

per row i. The Lasso problem (18) admits a closed-form solution, in terms of the *soft-thresholding* operator:

$$\mathbf{c}_{i,r+1} := \mathcal{S}(\boldsymbol{\alpha}_{i,r+1},\lambda_\tau/\rho) \qquad (19)$$

whose j-th entry is

$$c_{ij,r+1} = \begin{cases}|\alpha_{ij,r+1}|-\frac{\lambda_\tau}{\rho}\text{sign}(\alpha_{ij,r+1}), & \text{if } |\alpha_{ij,r+1}|>\frac{\lambda_\tau}{\rho}\\ 0, & \text{otherwise.}\end{cases} \qquad (20)$$

Algorithm 1 summarizes the steps outlined in this section for tracking the dynamic network topology. In practice, 5-10 inner ADMoM iterations suffice for convergence per $\tau = 1,\ldots,T$.

C. Algorithmic improvements

Online operation in delay-sensitive applications may not tolerate running multiple inner ADMoM iterations per time interval, while the matrix inversions in (12) incur $\mathcal{O}(N^3)$ complexity except when $\beta = 1$. These considerations motivate the following improvements to Algorithm 1:
I1. Single-iteration ADMoM, i.e., drop the **repeat** loop so that lines $6-16$ in Algorithm 1 are run once per time interval; and
I2. Leveraging the strict convexity of (6) w.r.t. $\mathbf{A}$ adopt the alternating-minimization algorithm (AMA) [9], which amounts to minimizing the *ordinary* Lagrangian w.r.t. $\mathbf{A}$ [instead of the augmented Lagrangian in (8a)]. This leads to updates $\tilde{\mathbf{a}}_{i,r+1} = (\mathbf{P}_i^\tau)^{-1}\mathbf{w}_i^\tau$, which can be recursively obtained with complexity $\mathcal{O}(N^2)$ using the matrix inversion lemma.

Due to space limitations, these algorithmic enhancements will be reported in a journal paper currently under preparation.

IV. Simulations

Numerical tests on synthetic network data are conducted here to evaluate the tracking ability of Algorithm 1. From a

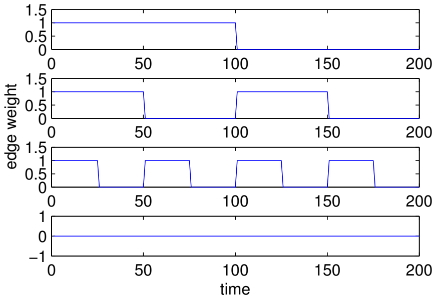

Fig. 2: Nonsmooth variation of edge weights.

"seed graph" with adjacency matrix

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

a Kronecker graph of size $N = 64$ nodes was generated as described in [5]. The resulting nonzero edge weights of $\mathbf{A}^t$ were allowed to vary over $T = 200$ intervals under 3 settings: i) i.i.d. Bernoulli(0.5) random variables; ii) random selection of the edge-evolution pattern uniformly from a set of 4 smooth functions: $a_{ij}(t) = 0.5 + 0.5\text{sin}(0.1t)$, $a_{ij}(t) = 0.5 + 0.5\text{cos}(0.1t)$, $a_{ij}(t) = e^{-0.01t}$, and $a_{ij}(t) = 0$; and iii) random selection of the edge-evolution pattern uniformly from a set of 4 nonsmooth functions shown in Fig. 2.

The number of contagions was set to $C = 80$, and $\mathbf{X}$ was formed with i.i.d. entries uniformly distributed over $[0, 3]$. Matrix $\mathbf{B}^t$ was set to diag($\mathbf{b}^t$), where $\mathbf{b}^t \in \mathbb{R}^N$ is a standard Gaussian random vector. During time interval t, infection times were generated synthetically as $\mathbf{Y}^t = (\mathbf{I}_N - \mathbf{A}^t)^{-1}(\mathbf{B}^t\mathbf{X} + \mathbf{E}^t)$, where $\mathbf{E}^t$ is a standard Gaussian random matrix.

With $\beta = 0.98$, $\rho = 1$, and $\epsilon = 10^{-3}$, Algorithm 1 was run after initializing $\mathbf{P}^0 = \mathbf{I}_N$, $\mathbf{Q}^0 = \mathbf{0}$, $\mathbf{\Gamma}_0 = \mathbf{0}$, $\mathbf{B}_0 = \text{diag}(\mathbf{1})$, $\mathbf{C}_0 = \mathbf{1}_{N\times N}$, $\mu^0 = 0$, and $\lambda_0 = 25$. In addition, $\lambda_\tau = \lambda_0$ for $\tau = 1, \ldots, T$. Fig. 3 shows the evolution of the mean-square error (MSE), $\sum_{i,j}(\hat{a}_{ij}^t - a_{ij}^t)^2/N^2$. The best performance was obtained when the temporal evolution of edges followed smooth functions. Although, the binary random evolution of edges resulted in the highest MSE, Algorithm 1 still tracked the underlying topology with reasonable accuracy as shown in the heat maps of the inferred adjacency matrices (Fig. 4). Non-smooth network evolution was tracked by both Algorithm 1 and one-step AMA. In addition to a significant speed-up in the online setting, comparable MSE was achieved by $t = 50$.

V. Concluding Summary

A dynamic SEM was proposed in this paper for network topology inference, using timestamp data for propagation of contagions typically observed in social networks. The model explicitly captures both topological influences and external sources of information diffusion over the unknown network. Exploiting the inherent edge sparsity typical of large networks, a novel algorithm based on ADMoM iterations was developed. Numerical tests demonstrate the potential of the approach for tracking the topology of slowly time-varying social networks.

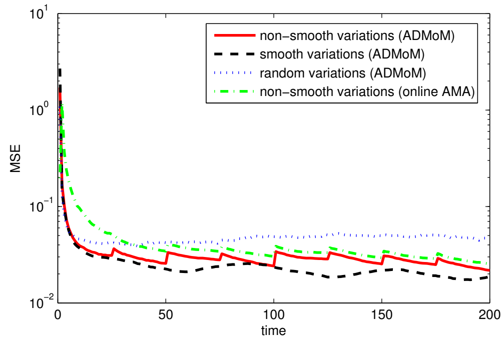

Fig. 3: Mean-square error evolution over time.

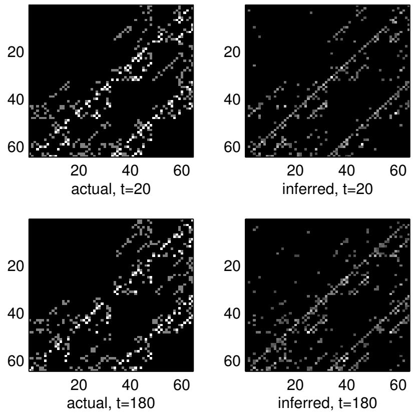

Fig. 4: Inferred and actual adjacency matrices at $t = 20, 180$.

Future research directions will focus on: i) developing first-order (e.g., stochastic gradient descent) algorithms for enhanced scalability; ii) studying convergence of the single-iteration ADMoM and AMA variants for topology tracking; and iii) carrying out tests with real network data such as traces of popular news items on the web.

References

[1] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm," *IEEE Trans. on Sig. Proc.*, vol. 58, pp. 3436–3447, July 2010.

[2] 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, pp. 1–122, July 2011.

[3] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Gene network inference via sparse structural equation modeling with genetic perturbations," *PLoS Comp. Biology*, vol. 9, e1003068, May 2013.

[4] Y. Chen, Y. Gu, and A. O. Hero III, "Sparse LMS for system identification," *Proc. of the Intl. Conf. on Acoust., Speech, and Signal Proc.,* Taipei, Taiwan, pp. 1952–1955, April 19–24, 2009.

[5] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: an approach to modeling networks," *J. Machine Learning Research*, vol. 11, pp. 985–1042, March 2010.

[6] S. Meyers and J. Leskovec, "On the convexity of latent social network inference," *Proc. of Neural Information Proc. Sys. Conf.*, Vancouver, Canada, Dec. 6–9, 2010.

[7] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," *Proc. of 28th Intl. Conf. Machine Learning*, Bellevue, Washington, USA, Jun. 28 – Jul. 2, 2011.

[8] M. G. Rodriguez, J. Leskovec, and B. Schölkopf, "Structure and dynamics of information pathways in online media," *Proc. of the 6th ACM Intl. Conf. on Web Search and Data Mining*, pp. 23–32, Feb. 2013.

[9] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim.*, vol. 29, pp. 119–138, Jan. 1991.