Dynamic Structural Equation Models for Tracking Cascades Over Social Networks

Brian Baingana, Gonzalo Mateos and Georgios B. Giannakis
Department of Electrical and Computer Engineering and Digital Technology Center
University of Minnesota, Minneapolis, MN 55455
{baing011,mate0058,georgios}@umn.edu

Abstract

Many real-world processes evolve in cascades over networks, whose topologies are often unobservable and change over time. However, the so-termed adoption times when for instance blogs mention popular news items are typically known, and are implicitly dependent on the underlying network. To infer the network topology, a *dynamic* structural equation model is adopted to capture the relationship between observed adoption times and the unknown edge weights. Assuming a slowly time-varying topology and leveraging the sparse connectivity inherent to social networks, edge weights are estimated by minimizing a sparsity-regularized exponentially-weighted least-squares criterion. To this end, a solver is developed by leveraging (pseudo) real-time sparsity-promoting proximal gradient iterations. Numerical tests with synthetic data and real cascades of online media demonstrate the effectiveness of the novel algorithm in unveiling sparse dynamically-evolving topologies, while accounting for external influences in the adoption times.

1 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 [28, 8]. For example, a terrorist attack may be reported within minutes on mainstream news websites. An information 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.

The propagation of a contagion is tantamount to *causal* effects or interactions being exerted among entities such as news portals and blogs, consumers, or people susceptible to being infected with a contagious disease. Acknowledging this viewpoint, *structural equation models* (SEMs) provide a general statistical modeling technique to estimate causal relationships among traits; see e.g., [12, 24]. These directional effects are often not revealed by standard linear models involving symmetric associations between random variables, such as those represented by covariances or correlations, [20], [9], [14]. SEMs are attractive because of their simplicity and ability to capture edge directionalities. They have been widely adopted in many fields, such as economics, psychometrics [22], social sciences [10], and recently in genetics for *static* gene regulatory network inference; see e.g., [5, 18] and references therein. However, SEMs have not been utilized to track the dynamics of causal effects among interacting nodes. In this context, the present paper proposes a *dynamic* SEM to account for time-varying directed networks over which contagions propagate, and describes how node infection times depend on both topological and external influences. 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. External influence data is also useful for model identifiability, and has been shown necessary to resolve directional ambiguities [3].

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 (Section 3). 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 [6, 15, 1]. A novel algorithm to jointly track the network's adjacency matrix and the weights capturing the level of external influences is developed in Section 3.1, which minimizes the resulting non-differentiable cost function via a proximal-gradient (PG) solver; see e.g., [23, 7, 4]. The resulting dynamic iterative shrinkage-thresholding algorithm (ISTA) is provably convergent, and offers parallel, closed-form, and sparsity-promoting updates per iteration. Numerical tests on synthetic network data demonstrate the superior error performance of the developed algorithms, and highlight their merits when compared to the sparsity-agnostic approach in [27]. Experiments in Section 4 involve real temporal traces of popular global events that propagated on news websites and blogs in 2011 [17]. Interestingly, topologies inferred from cascades associated to the meme "Kim Jong-un" exhibit an abrupt increase in the number of edges following the appointment of the new North Korean ruler.

Related work. Inference of networks using temporal traces of infection events has recently become an active area of research. According to the taxonomy in [13, Ch. 7], this can be viewed as a problem involving inference of *association* networks. Several prior approaches postulate probabilistic models and rely on maximum likelihood estimation (MLE) to infer edge weights as pairwise transmission rates between nodes [26], [21]. 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 [27]. Although successful experiments on large-scale web data reliably uncover information pathways, the estimator in [27] 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.

Notation. Bold uppercase (lowercase) letters will denote matrices (column vectors), while operators $(\cdot)^\top$, $\lambda_{\max}(\cdot)$, and $\text{diag}(\cdot)$ will stand for matrix transposition, spectral radius, and diagonal matrix, respectively. The $N \times N$ identity matrix will be represented by $\mathbf{I}_N$, while $\mathbf{0}_N$ will denote the $N \times 1$ vector of all zeros, and $\mathbf{0}_{N\times P} := \mathbf{0}_N \mathbf{0}_P^\top$. The ℓ_p and Frobenius norms will be denoted by $\|\cdot\|_p$, and $\|\cdot\|_F$, respectively.

2 Network Model and Problem Statement

Consider a dynamic network with N nodes observed over time intervals $t = 1, \ldots, T$, whose abstraction is a graph with topology described by an unknown, time-varying, and weighted adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{N\times N}$. Entry (i, j) of $\mathbf{A}^t$ (henceforth denoted by a_{ij}^t) is nonzero only if a directed edge connects nodes i and j (pointing from j to i) during the time interval t. As a result, one in general has $a_{ij}^t \neq a_{ji}^t$, i.e., matrix $\mathbf{A}^t$ is generally non-symmetric, which is suitable to model directed networks. Note that the model tacitly assumes that the network topology remains fixed during any given time interval t, but can change across time intervals.

Suppose C contagions propagate over the network, and the infection time of node i by contagion c 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.

The infection time of node i during interval t is modeled according to the following *dynamic* structural equation model (SEM)

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

where b_{ii}^t captures the time-varying level of influence of external sources, and e_{ic}^t accounts for measurement errors and unmodeled dynamics. It follows from (1) that if $a_{ij}^t \neq 0$, then y_{ic}^t is affected by the value of y_{jc}^t. With $\mathbf{B}^t := \text{diag}(b_{11}, \ldots, b_{NN})$, collecting observations for the entire network and all C contagions yields the dynamic matrix SEM

$$\mathbf{Y}^t = \mathbf{A}^t\mathbf{Y}^t + \mathbf{B}^t\mathbf{X} + \mathbf{E}^t \tag{2}$$

where $\mathbf{Y}^t := [y_{ic}^t]$, $\mathbf{X} := [x_{ic}]$, and $\mathbf{E}^t := [e_{ic}^t]$ are all $N \times C$ matrices. A single network topology $\mathbf{A}^t$ is adopted for all contagions, which is suitable e.g., when information cascades are formed around a common meme or trending (news) topic in the Internet; see also the data in Section 4.

Given $\{\mathbf{Y}^t\}_{t=1}^T$ and $\mathbf{X}$ adhering to (2), 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 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.

3 Topology Tracking Algorithm

To estimate $\{\mathbf{A}^t, \mathbf{B}^t\}$ in a static setting, one can solve the following regularized LS problem

$$\begin{aligned} \{\hat{\mathbf{A}}, \hat{\mathbf{B}}\} = \arg\min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 + \lambda\|\mathbf{A}\|_1 \\ \text{s. to} \quad & a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j \end{aligned} \tag{3}$$

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 $a_{ii} = 0$, while having $b_{ij} = 0,\ \forall i \neq j$, ensures that $\hat{\mathbf{B}}$ is diagonal as in (2). The batch estimator (3) yields single estimates $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}\}$ that best fit the data $\{\mathbf{Y}^t\}_{t=1}^T$ and $\mathbf{X}$ over the whole measurement horizon $t = 1, \ldots, T$, and as such (3) neglects potential network variations across time intervals.

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 minimum storage requirements. Recursive solvers enabling sequential inference of the underlying dynamic network topology are thus preferred.

For $t = 1, \ldots, T$, consider the sparsity-regularized exponentially-weighted LS estimator (EWLSE)

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

where $\beta \in (0, 1]$ is the forgetting factor that forms estimates $\{\hat{\mathbf{A}}^t, \hat{\mathbf{B}}^t\}$ using all measurements acquired until time t. Whenever $\beta < 1$, past data are exponentially discarded thus enabling tracking of dynamic network topologies. The first summand in the cost corresponds to an exponentially-weighted moving average (EWMA) of the squared model residuals norms. The EWMA can be seen as an average modulated by a sliding window of equivalent length $1/(1 - \beta)$, which clearly grows as $\beta \to 1$. In the infinite-memory setting whereby $\beta = 1$, (4) boils down to the batch estimator (3).

Selection of the (possibly time-varying) tuning parameter λ_t is an important aspect of regularization methods such as (4), because λ_t controls the sparsity level of the inferred network and how its structure may change over time. For sufficiently large values of λ_t one obtains the trivial solution $\hat{\mathbf{A}}^t = \mathbf{O}_{N\times N}$, while increasingly more dense graphs are obtained as $\lambda_t \to 0$. An increasing λ_t will be required for accurate estimation over extended time-horizons, since for $\beta \approx 1$ the norm of the LS term in (4) grows due to noise accumulation. This way the effect of the regularization term will be downweighted unless one increases λ_t at a suitable rate, for instance proportional to $\sqrt{\sigma^2 t}$ as suggested by large deviation tail bounds when the errors are assumed $e_{ic}^t \sim \mathcal{N}(0, \sigma^2)$, and the problem dimensions N, C, T are sufficiently large [20, 19, 1]. In the topology tracking experiments of Section 4, a time-invariant value of λ is adopted and typically chosen via trial and error to optimize the performance. This is justified since smaller values of β are selected for tracking network variations, which also implies that past data (and noise) are discarded faster, and the norm of the LS term in (4) remains almost invariant. As future research it would be interesting to delve further into the choice of λ_t using model selection techniques such as cross-validation [5], Bayesian information criterion (BIC) scores [14], or the minimum description length (MDL) principle [25], and investigate how this choice relates to statistical model consistency in a dynamic setting.

3.1 Proximal gradient algorithm

Proximal gradient (PG) algorithms have been popularized for ℓ_1-norm regularized linear regression problems, through the class of iterative shrinkage-thresholding algorithms (ISTA); see e.g., [7] and [23] for a comprehensive tutorial treatment. The main advantage of ISTA over off-the-shelf interior point methods is its computational simplicity. Iterations boil down to matrix-vector multiplications involving the regression matrix, followed by a soft-thresholding operation [11, p. 93].

In the sequel, an ISTA algorithm is developed for the sparsity regularized dynamic SEM formulation (4) at time t. Based on this module, a (pseudo) real-time algorithm for tracking the dynamically-evolving network topology over the horizon $t = 1, \ldots, T$ is obtained as well. The algorithm's memory storage requirement and computational cost per sample $\{\mathbf{Y}^t, \mathbf{X}\}$ does not grow with t.

Solving (4) **for a single time interval** t**.** Introducing the optimization variable $\mathbf{V} := [\mathbf{A}\,\mathbf{B}]$, observe that the gradient of $f(\mathbf{V}) := \frac{1}{2}\sum_{\tau=1}^{t}\beta^{t-\tau}\|\mathbf{Y}^\tau - \mathbf{A}\mathbf{Y}^\tau - \mathbf{B}\mathbf{X}\|_F^2$ is Lipschitz continuous with a Lipschitz constant $L_f = \lambda_{\max}(\sum_{\tau=1}^{t}\beta^{t-\tau}[(\mathbf{Y}^\tau)^\top \ \ (\mathbf{X})^\top]^\top[(\mathbf{Y}^\tau)^\top \ \ (\mathbf{X})^\top])$, i.e., $\|\nabla f(\mathbf{V}_1) - \nabla f(\mathbf{V}_2)\| \leq L_f\|\mathbf{V}_1 - \mathbf{V}_2\|, \forall\, \mathbf{V}_1,\ \mathbf{V}_2$ in the domain of f. The Lipschitz constant is time varying, but the dependency on t is kept implicit for notational convenience. Instead of directly optimizing the cost in (4), PG algorithms minimize a sequence of overestimators evaluated at judiciously chosen points (typically the current iterate, or a linear combination of the two previous iterates).

With $k = 1, 2, \ldots$ denoting iterations and upon defining $g(\mathbf{V}) := \lambda_t\|\mathbf{A}\|_1$, PG algorithms iterate

$$\mathbf{V}[k] := \arg\min_{\mathbf{V}} \left\{ \frac{L_f}{2}\|\mathbf{V} - \mathbf{G}(\mathbf{V}[k-1])\|_F^2 + g(\mathbf{V}) \right\} \tag{5}$$

where $\mathbf{G}(\mathbf{V}[k-1]) := \mathbf{V}[k-1] - (1/L_f)\nabla f(\mathbf{V}[k-1])$ corresponds to a gradient-descent step taken from $\mathbf{V}[k-1]$, with step-size equal to $1/L_f$. The optimization problem (5) is known as the *proximal operator* of the function g/L_f evaluated at $\mathbf{G}(\mathbf{V}[k-1])$, and is denoted as $\text{prox}_{g/L_f}(\mathbf{G}(\mathbf{V}[k-1]))$. Henceforth adopting the notation $\mathbf{G}[k-1] := \mathbf{G}(\mathbf{V}[k-1])$ for convenience, the PG iterations can be compactly rewritten as $\mathbf{V}[k] = \text{prox}_{g/L_f}(\mathbf{G}[k-1])$.

A key element to the success of PG algorithms stems from the possibility of efficiently evaluating the proximal operator (cf. (5)). Specializing to (4), note that (5) decomposes into

$$\mathbf{A}[k] := \arg\min_{\mathbf{A}} \left\{ \frac{L_f}{2}\|\mathbf{A} - \mathbf{G}_A[k-1]\|_F^2 + \lambda_t\|\mathbf{A}\|_1 \right\} = \mathcal{S}_{\lambda_t/L_f}(\mathbf{G}_A[k-1]) \tag{6}$$

$$\mathbf{B}[k] := \arg\min_{\mathbf{B}} \left\{ \|\mathbf{B} - \mathbf{G}_B[k-1]\|_F^2 \right\} = \mathbf{G}_B[k-1] \tag{7}$$

subject to the constraints in (4) which so far have been left implicit, and $\mathbf{G} := [\mathbf{G}_A\ \mathbf{G}_B]$. Letting $\mathcal{S}_\mu(\mathbf{M})$ with (i,j)-th entry given by $\text{sign}(m_{ij})\max(|m_{ij}| - \mu, 0)$ denote the soft-thresholding operator, it follows that $\text{prox}_{\lambda_t\|\cdot\|_1/L_f}(\cdot) = \mathcal{S}_{\lambda_t/L_f}(\cdot)$, e.g., [7, 11]. Because there is no regularization on the matrix $\mathbf{B}$, the corresponding update (7) boils-down to a simple gradient-descent step.

What remains now is to obtain expressions for the gradient of $f(\mathbf{V})$ with respect to $\mathbf{A}$ and $\mathbf{B}$, which are required to form the matrices $\mathbf{G}_A$ and $\mathbf{G}_B$. To this end, note that by incorporating the constraints $a_{ii} = 0$ and $b_{ij} = 0,\ \forall j \neq i, i = 1, \ldots N$, one can simplify the expression of $f(\mathbf{V})$ as

$$f(\mathbf{V}) := \frac{1}{2}\sum_{\tau=1}^{t}\sum_{i=1}^{N}\beta^{t-\tau}\|(\mathbf{y}_i^\tau)^\top - \mathbf{a}_{-i}^\top\mathbf{Y}_{-i}^\tau - b_{ii}\mathbf{x}_i^\top\|_F^2 \tag{8}$$

where $(\mathbf{y}_i^\tau)^\top$ and $\mathbf{x}_i^\top$ denote the i-th row of $\mathbf{Y}^\tau$ and $\mathbf{X}$, respectively; while $\mathbf{a}_{-i}^\top$ denotes the $1 \times (N-1)$ vector obtained by removing entry i from the i-th row of $\mathbf{A}$, and likewise $\mathbf{Y}_{-i}^\tau$ is the $(N-1) \times C$ matrix obtained by removing row i from $\mathbf{Y}^\tau$. It is apparent from (8) that $f(\mathbf{V})$ is separable across the trimmed row vectors $\mathbf{a}_{-i}^{\top}$, and the diagonal entries b_{ii}, $i = 1, \ldots, N$. The sought gradients are

$$\nabla_{\mathbf{a}_{-i}} f(\mathbf{V}) = \mathbf{\Sigma}_{-i}^{t} \mathbf{a}_{-i} + \bar{\mathbf{Y}}_{-i}^{t} \mathbf{x}_i b_{ii} - \boldsymbol{\sigma}_{-i}^{t} \tag{9}$$

$$\nabla_{b_{ii}} f(\mathbf{V}) = \mathbf{a}_{-i}^{\top} \bar{\mathbf{Y}}_{-i}^{t} \mathbf{x}_i + \frac{1-\beta^t}{1-\beta} b_{ii} \|\mathbf{x}_i\|_2^2 - (\bar{\mathbf{y}}_i^{\tau})^{\top} \mathbf{x}_i. \tag{10}$$

where $(\bar{\mathbf{y}}_i^t)^{\top}$ denotes the i-th row of $\bar{\mathbf{Y}}^t := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{Y}^{\tau}$, and $\bar{\mathbf{Y}}_{-i}^{t} := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{Y}_{-i}^{\tau}$. Similarly, $\boldsymbol{\sigma}_{-i}^{t} := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{Y}_{-i}^{\tau} \mathbf{y}_i^{\tau}$ and $\mathbf{\Sigma}_{-i}^{t}$ is obtained by removing the i-th row and i-th column from $\mathbf{\Sigma}^t := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{Y}^{\tau} (\mathbf{Y}^{\tau})^{\top}$. From (6)-(7) and (9)-(10), the parallel ISTA iterations

$$\nabla_{\mathbf{a}_{-i}} f[k] = \mathbf{\Sigma}_{-i}^{t} \mathbf{a}_{-i}[k] + \bar{\mathbf{Y}}_{-i}^{t} \mathbf{x}_i b_{ii}[k] - \boldsymbol{\sigma}_{-i}^{t} \tag{11}$$

$$\nabla_{b_{ii}} f[k] = \mathbf{a}_{-i}^{\top}[k] \bar{\mathbf{Y}}_{-i}^{t} \mathbf{x}_i + \frac{(1-\beta^t)}{1-\beta} b_{ii}[k] \|\mathbf{x}_i\|_2^2 - (\bar{\mathbf{y}}_i^{t})^{\top} \mathbf{x}_i \tag{12}$$

$$\mathbf{a}_{-i}[k+1] = \mathcal{S}_{\lambda_t / L_f} \left(\mathbf{a}_{-i}[k] - (1/L_f) \nabla_{\mathbf{a}_{-i}} f[k] \right) \tag{13}$$

$$b_{ii}[k+1] = b_{ii}[k] - (1/L_f) \nabla_{b_{ii}} f[k] \tag{14}$$

are provably convergent to the globally optimal solution $\{\hat{\mathbf{A}}^t, \hat{\mathbf{B}}^t\}$ of (4), as per the general convergence results available for PG methods and ISTA in particular [7, 23].

Computation of the gradients in (11)-(12) requires one matrix-vector mutiplication by $\mathbf{\Sigma}_{-i}^{t}$ and one by $\bar{\mathbf{Y}}_{-i}^{t}$, in addition to three vector inner-products, plus a few (negligibly complex) scalar and vector additions. Both the update of $b_{ii}[k+1]$ as well as the soft-thresholding operation in (13) entail negligible computational complexity. All in all, the simplicity of the resulting iterations should be apparent. Per iteration, the actual rows of the adjacency matrix are obtained by zero-padding the updated $\mathbf{a}_{-i}[k]$, namely setting

$$\mathbf{a}_i^{\top}[k] = [a_{-i,1}[k] \ldots a_{-i,i-1}[k] \; 0 \; a_{-i,i}[k] \ldots a_{-i,N}[k]]. \tag{15}$$

This way, the desired SEM parameter estimates at time t are given by $\hat{\mathbf{A}}^t = [\mathbf{a}_1^{\top}[k], \ldots, \mathbf{a}_N^{\top}[k]]^{\top}$ and $\hat{\mathbf{B}}^t = \text{diag}(b_{11}[k], \ldots, b_{NN}[k])$, for k large enough so that convergence has been attained.

Solving (4) **over the entire time horizon** $t = 1, \ldots, T$**.** To track the dynamically-evolving network topology, one can go ahead and solve (4) sequentially for each $t = 1, \ldots, T$ as data arrive, using (11)-(14). (The procedure can also be adopted in a batch setting, when all $\{\mathbf{Y}^t\}_{t=1}^{T}$ are available in memory.) Because the network is assumed to vary slowly across time, it is convenient to warm-restart the ISTA iterations, that is, at time t initialize $\{\mathbf{A}[0], \mathbf{B}[0]\}$ with the solution $\{\hat{\mathbf{A}}^{t-1}, \hat{\mathbf{B}}^{t-1}\}$. This way, for smooth network variations one expects convergence to be attained after few iterations.

To obtain the new SEM parameter estimates via (11)-(14), it suffices to update (possibly) λ_t and the Lipschitz constant L_f, as well as the data-dependent EWMAs $\mathbf{\Sigma}^t$, and $\bar{\mathbf{Y}}^t$. Interestingly, the potential growing-memory problem in storing the entire history of data $\{\mathbf{Y}^t\}_{t=1}^{T}$ can be avoided by performing the recursive updates

$$\mathbf{\Sigma}^t = \beta \mathbf{\Sigma}^{t-1} + \mathbf{Y}^t (\mathbf{Y}^t)^{\top}, \quad \bar{\mathbf{Y}}^t = \beta \bar{\mathbf{Y}}^{t-1} + \mathbf{Y}^t. \tag{16}$$

The complexity in evaluating the Gram matrix $\mathbf{Y}^t (\mathbf{Y}^t)^{\top}$ dominates the per-iteration computational cost of the algorithm. To circumvent the need of recomputing the Lipschitz constant per time interval, the step-size $1/L_f$ in (13)-(14) can be selected by a line search [23]. One choice is the backtracking step-size rule [4], for which convergence to $\{\hat{\mathbf{A}}^t, \hat{\mathbf{B}}^t\}$ can be established as well.

Algorithm 1 summarizes the steps outlined in this section for tracking the dynamic network topology, given temporal traces of infection events $\{\mathbf{Y}^t\}_{t=1}^{T}$ and susceptibilities $\mathbf{X}$. It is termed *pseudo real-time* ISTA, since in principle one needs to run multiple (inner) ISTA iterations till convergence per time interval $t = 1, \ldots, T$. This will in turn incur an associated delay, that may (or may not) be tolerable depending on the specific network inference problem at hand. Nevertheless, numerical tests indicate that in practice 5-10 inner iterations suffice for convergence; see also the extended journal version of this paper [2].

4 Numerical Tests

Performance of Algorithm 1 is assessed in this section via simulations using both synthetically-generated network data, and real traces of information cascades collected from the web [17].

Algorithm 1 Pseudo real-time ISTA for topology tracking

```
Require: $\{\mathbf{Y}^t\}_{t=1}^T$, $\mathbf{X}$, $\beta$.
Initialize $\hat{\mathbf{A}}^0 = \mathbf{0}_{N\times N}, \hat{\mathbf{B}}^0 = \mathbf{\Sigma}^0 = \mathbf{I}_N, \bar{\mathbf{Y}}^0 = \mathbf{0}_{N\times C}, \lambda_0$.
for $t = 1, \ldots, T$ do
    Update $\lambda_t$, $L_f$ and $\mathbf{\Sigma}^t$, $\bar{\mathbf{Y}}^t$ via (16).
    Initialize $\mathbf{A}[0] = \hat{\mathbf{A}}^{t-1}$, $\mathbf{B}[0] = \hat{\mathbf{B}}^{t-1}$, and set $k = 0$.
    while not converged do
        for $i = 1 \ldots N$ (in parallel) do
            Compute $\mathbf{\Sigma}^t_{-i}$ and $\bar{\mathbf{Y}}^t_{-i}$.
            Form gradients at $\mathbf{a}_{-i}[k]$ and $b_{ii}[k]$ via (11)-(12).
            Update $\mathbf{a}_{-i}[k+1]$ via (13).
            Update $b_{ii}[k+1]$ via (14).
            Update $\mathbf{a}_i[k+1]$ via (15).
        end for
        $k = k + 1$.
    end while
    return $\hat{\mathbf{A}}^t = \mathbf{A}[k]$, $\hat{\mathbf{B}}^t = \mathbf{B}[k]$.
end for
```

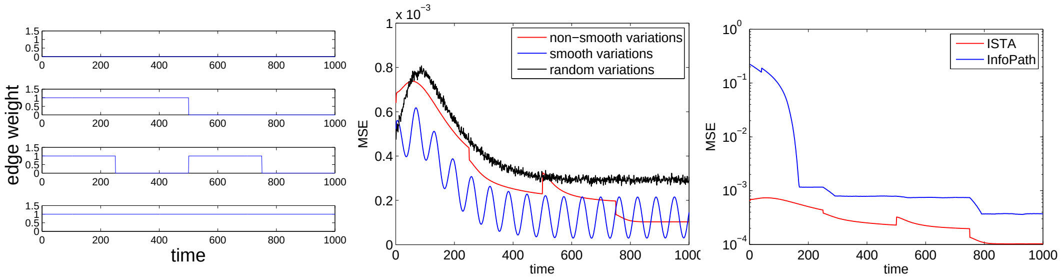


Figure 1: (Left) Nonsmooth variation of synthetically-generated edge weights of the time-varying network. (Center) MSE versus time for different edge evolution patterns. (Right) MSE performance comparison between Algorithm 1 and InfoPath [27].

Synthetic data generation. Numerical tests on synthetic network data are conducted here to evaluate the tracking ability of Algorithm 1 [1]. A random graph of $N = 100$ nodes was generated using the Barabási-Albert model [16] allowing each new node to attach itself to at most 2 existing nodes. The resulting nonzero edge weights of $\mathbf{A}^t$ were allowed to vary over $T = 1{,}000$ 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 four 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 four nonsmooth functions shown in Fig. 1 (left).

The number of contagions was set to $C = 150$, 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.

Performance evaluation. With $\beta = 0.98$, Algorithm 1 was run after initializing the relevant variables as described in the algorithm table (cf. Section 3.1), and setting $\lambda_0 = 140$. In addition, $\lambda_t = \lambda_0$ for $t = 1, \ldots, T$ as discussed in Section 3. Fig. 1 (center) shows the evolution of the mean-square error (MSE), $\sum_{i,j}(\hat{a}^t_{ij} - a^t_{ij})^2/N^2$. As expected, the best performance was obtained when the temporal evolution of edges followed smooth functions. Even though the Bernoulli evolution of edges resulted in the highest MSE, Algorithm 1 still tracked the underlying topology with reasonable accuracy as depicted in the heat maps of the inferred adjacency matrices; see Fig. 2 (left).

[1] The Matlab implementation of Algorithm 1 used here can handle networks of several thousand nodes. Still a smaller network is analyzed since results are still representative of the general behavior, and offers better visualization of the results in e.g., the adjacency matrices in Fig. 2.

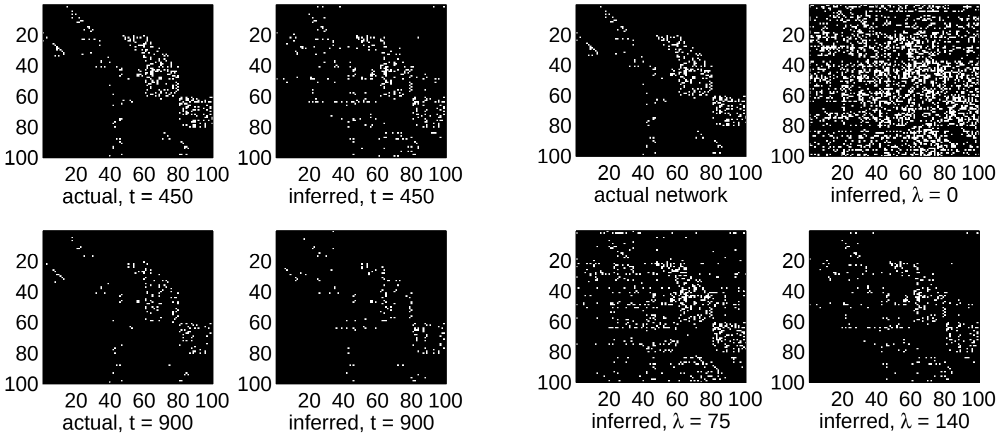


Figure 2: (Left) Actual adjacency matrix $\mathbf{A}^t$ and corresponding estimate $\hat{\mathbf{A}}^t$ obtained at time intervals $t = 450$ and $t = 900$. (Right) Actual adjacency matrix at $t = 450$ compared with the inferred adjacency matrices with $\lambda_t = \lambda$ for all t and $\lambda = 0$, $\lambda = 75$, and $\lambda = 140$.

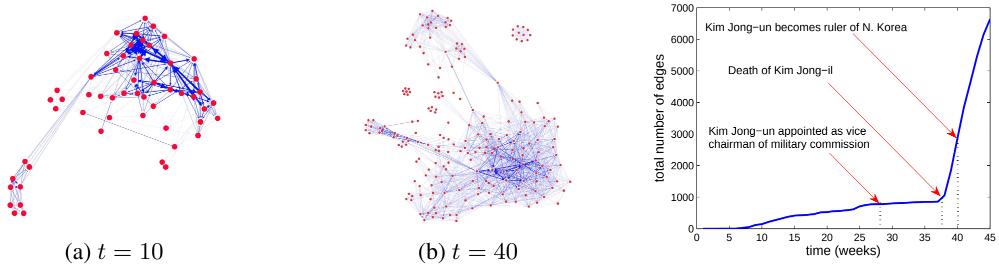
To illustrate the importance of leveraging sparsity of the edge weights, Fig. 2 (right) depicts heatmaps of the adjacency matrices inferred at $t = 450$, with λ set to $0, 75$, and 140 for all time intervals. Comparisons with the actual adjacency matrix reveal that increasing λ progressively refines the network estimates by driving erroneously detected nonzero edge weights to 0. Indeed, the value $\lambda = 140$ in this case appears to be just about right, while smaller values markedly overestimate the support set associated with the edges present in the actual network.

Comparison with [27]. Algorithm 1 is compared here to the method of [27], which does not explicitly account for external influences and edge sparsity. To this end, the stochastic-gradient descent algorithm (a.k.a. "InfoPath") developed in [27] is run using the generated synthetic data with non-smooth edge variations. Postulating an exponential transmission model, the dynamic network is tracked by InfoPath by performing MLE of the edge transmission rates (see [27] for details of the model and the algorithm). Note that the postulated model therein differs from (2), used here to generate the network data. Fig. 1 (right) depicts the MSE performance of "InfoPath" compared against ISTA. Apparently, there is an order of magnitude reduction in MSE by explicitly modeling external sources of influence and leveraging the attribute of sparsity.

Real dataset description. The real data used was collected during a prior study by monitoring blog posts and news articles for memes (popular textual phrases) appearing within a set of over 3.3 million websites [27]. Traces of information cascades were recorded over a period of one year, from March 2011 till February 2012; the data is publicly available from [17]. The time when each website mentioned a specific news item was recorded as a Unix timestamp in hours (i.e., the number of hours since midnight on January 1, 1970). Specific globally-popular topics during this period were identified and cascade data for the top $5,000$ websites that mentioned memes associated with them were retained. The real-data tests that follow focus on the topic "Kim Jong-un", the current leader of North Korea whose popularity rose after the death of his father and predecessor, during the observation period.

Data was first pre-processed and filtered so that only (significant) cascades that propagated to at least 7 websites were retained. This reduced the dataset significantly to the 360 most relevant websites over which 466 cascades related to "Kim Jong-un" propagated. The observation period was then split into $T = 45$ weeks, and each time interval was set to one week. In addition, the observation time-scale was adjusted to start at the beginning of the earliest cascade.

Matrix $\mathbf{Y}^t$ was constructed by setting y_{ic}^t to the time when website i mentioned phrase c if this occurred during the span of week t. Otherwise y_{ic}^t was set to a large number, $100t_{\text{max}}$, where t_{max} denotes the largest timestamp in the dataset. Typically the entries of matrix $\mathbf{X}$ capture prior knowledge about the susceptibility of each node to each contagion. For instance, the entry $\mathbf{x}_{ic}$ could denote the online search rank of website i for a search keyword associated with contagion c. In the



(a) $t = 10$

(b) $t = 40$

Figure 3: (Left) Visualization of the estimated networks obtained by tracking those information cascades related to the topic “Kim Jong-un”. (Right) Evolution of total number of inferred edges.

absence of such real data, the entries of $\mathbf{X}$ were generated randomly from a uniform distribution over the interval $[0, 0.01]$.

Experimental results. Algorithm 1 was run on real data with $\beta = 0.9$ and $\lambda_t = 100$. Fig. 3 (left) depicts drawings of the inferred network at $t = 10$ and $t = 40$ weeks. Little was known about Kim Jong-un during the first 10 weeks of the observation period. However, speculation about the possible successor of the dying North Korean ruler, Kim Jong-il, rose until his death on December 17, 2011 (week 38). He was succeeded by Kim Jong-un on December 30, 2011 (week 40). The network visualizations show an increasing number of edges over the 45 weeks, illustrating the growing interest of international news websites and blogs in the new ruler. Unfortunately, the observation horizon does not go beyond $T = 45$ weeks. A longer span of data would have been useful to investigate at what rate did the global news coverage on the topic eventually subside.

Fig. 3 (right) depicts the time evolution of the total number of edges in the inferred dynamic network. Of particular interest are the weeks during which: i) Kim Jong-un was appointed as the vice chairman of the North Korean military commission; ii) Kim Jong-il died; and iii) Kim Jong-un became the ruler of North Korea. These events were the topics of many online news articles and political blogs, an observation that is reinforced by the experimental results shown in the plot.

5 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 computationally-efficient proximal gradient algorithm with well-appreciated convergence properties was developed to minimize a suitable sparsity-regularized exponentially-weighted LS estimator. A number of experiments conducted on both synthetically-generated as well as real data demonstrated the effectiveness of the proposed algorithms in tracking dynamic and sparse networks.

The present work opens up multiple directions for exciting follow-up research. Future and ongoing research includes: i) investigating the conditions for identifiability of sparse and dynamic SEMs, as well as their statistical consistency properties tied to the selection of λ_t; ii) generalizing the SEM using kernels or suitable graph similarity measures to capture nonlinear dependencies and also enable network topology forecasting; and iii) exploiting the algorithm’s parallel structure to devise MapReduce/Hadoop implementations scalable to million-node graphs.

Acknowledgments

Work in this paper was supported by the NSF ECCS Grants No. 1202135 and No. 1343248, and the NSF AST Grant No. 1247885.

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. Signal Process.*, vol. 58, pp. 3436–3447, Jul. 2010.

[2] B. Baingana, G. Mateos, and G. B. Giannakis, “Dynamic structural equation models for social network topology inference,” *IEEE J. Selected Topics in Signal Process.*, Aug. 2013 (submitted; see also arXiv:1309.6683v2 [cs.SI]).

[3] 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.

[4] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, Jan. 2009.

[5] 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, May 2013.

[6] Y. Chen, Y. Gu, and A. O. Hero III, “Sparse LMS for system identification,” in *Proc. of Intern. Conf. on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009.

[7] I. Daubechies, M. Defrise, and C. D. Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, Aug. 2004.

[8] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. New York, NY: Cambridge University Press, 2010.

[9] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, pp. 432–441, Dec. 2007.

[10] A. S. Goldberger, “Structural equation methods in the social sciences,” *Econometrica*, vol. 40, pp. 979–1001, Nov. 1972.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.

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

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

[14] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, “Estimating time-varying networks,” *Ann. Appl. Statist.*, vol. 4, pp. 94–123, 2010.

[15] Y. Kopsinis, K. Slavakis, and S. Theodoridis, “Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls,” *IEEE Trans. Signal Process.*, vol. 59, pp. 936–952, Mar. 2011.

[16] B. Albert-László and A. Réka, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, pp. 509–512, Oct. 1999.

[17] J. Leskovec, “Web and blog datasets,” *Stanford Network Analysis Project*, 2011. [Online]. Available: http://snap.stanford.edu/infopath/data.html

[18] B. A. Logsdon and J. Mezey, “Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations,” *PLoS Comp. Biology*, vol. 6, Dec. 2010.

[19] M. Mardani, G. Mateos, and G. B. Giannakis, “Dynamic anomalography: Tracking network anomalies via sparsity and low rank,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, pp. 50–66, Feb. 2013.

[20] N. Meinshausen and P. Buhlmann, “High-dimensional graphs and variable selection with the lasso,” *Ann. Statist.*, vol. 34, pp. 1436–1462, 2006.

[21] S. Meyers and J. Leskovec, “On the convexity of latent social network inference,” in *Proc. of Neural Information Proc. Sys. Conf.*, Vancouver, Canada, Feb. 2013.

[22] B. Muthén, “A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators,” *Pyschometrika*, vol. 49, pp. 115–132, Mar. 1984.

[23] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optimization*, vol. 1, pp. 123–231, 2013.

[24] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge University Press, 2009.

[25] I. Ramirez and G. Sapiro, “An MDL framework for sparse coding and dictionary learning,” *IEEE Trans. Signal Process.*, vol. 60, pp. 2913–2927, Jun. 2012.

[26] M. G. Rodriguez, D. Balduzzi, and B. Scholkopf, “Uncovering the temporal dynamics of diffusion networks,” in *Proc. of 28th Intern. Conf. Machine Learning*, Bellevue, WA, Jul. 2011.

[27] M. G. Rodriguez, J. Leskovec, and B. Scholkopf, “Structure and dynamics of information pathways in online media,” in *Proc. of 6th ACM Intern. Conf. on Web Search and Data Mining*, Rome, Italy, Dec. 2010.

[28] E. M. Rogers, *Diffusion of Innovations*, 4th ed. Free Press, 1995.